

24 SEASON DRUMS SOCIETY
MANAGEMENT SYSTEM

SUM YONG TAT

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING)

UNIVERSITY MALAYSIA PAHANG

SUPERVISOR'S DECLARATION

“I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Sciences (Software Engineering)”

Signature :

Name of Supervisor : ROZLINA BINTI MOHAMED

Date : 21 MAY 2012

STUDENT'S DECLARATION

I declare that this thesis entitled “24 Season Drums Society Management System” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : SUM YONG TAT

Date : 21 MAY 2012

DEDICATION

*Special dedication to my beloved family,
my lecturer, my fellow friends,
and all faculty members.*

For all your support and believe in me.

Sincerely,

SUM YONG TAT

ACKNOWLEDGEMENT

I am grateful and would like to express my sincere gratitude to my supervisor Miss Rozlina Binti Mohamed for her great ideas and invaluable guidance in making this project possible. She also gives me continuous encouragement and constant support while doing the project. I appreciate her hard work to guide me from the first day I approach to her to have her as my supervisor to these concluding moments. I am truly grateful for her progressive vision about my training in writing good documentation and she often impressed me with creative idea in designing the system flow. I often make naïve mistakes but she able to tolerance with me and lead me back to the correct way.

My sincere thanks go to the lecturers of my faculty who helped me directly or indirectly to complete this project. I also would like to express a very special thanks to the evaluators of my project for their suggestions and opinions. I also sincerely thanks to all my friends in the university who helped me in many ways and their excellent co-operation, inspiration and supports.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals and studies. They consistently encouraged me to carry on my studies and make achievements in life.

ABSTRACT

24 Season Drums Society Management System is an online base management system to automate the managing process of society. The current system of the society is old traditional filing system to record the member profile, resources reservation, and attendance list and do not have a proper management system used and lack of proper platform for support and problem solving solution of member and open users. Therefore this system is develop for automate the managing process of society. There are three target users in this system, which are the administrator, society member and open users. This system has total of twenty main modules in this system, which is Manage donation, resource booking, resource availability, posting news and events, add comments to news & events, manual performance/show timetable, picture gallery, Search on photo, attendance management, management for society account, video gallery, search on video, manage members profile, request for show and performance, performance management, society complaint management, help and information, self timetable, resource/asset management and portal management. Adopting V-model in iterative and incremental development process is used to develop this project. There are separate to 2 phases which are verification phases and validation phases. In verification phases have requirement analysis, system design architecture design and coding. Validation phases have unit testing, integration testing, system testing and acceptance testing. The twenty modules will be divided to various independent parts by incremental development which is 6 divisions, the priority service of these incremental set also arranged in the order which the first set is the highest priority services. This system succeeds to achieve the objectives and solve the problems stated earlier.

ABSTRAK

24 Season Drums Society Management System adalah sistem pengurusan asas untuk megautomasikan proses urusan society. Society management system adalah system pemfailan tradisional yang lama untuk merekodkan profil ahli, tempahan sumber-sumber, dan senarai kehadiran dan tidak mempunyai sistem pengurusan yang sesuai digunakan dan kekurangan platform yang betul bagi sokongan dan penyelesaian masalah ahli dan pengguna terbuka. Oleh itu sistem ini dibangunkan untuk mengautomaikan proses pengurusan society. Terdapat tiga pengguna sasaran dalam sistem ini, iaitu pentadbir, ahli society and pengguna terbuka. Sistem ini mempunyai jumlah dua puluh modul utama, iaitu mengurus derma, tempahan sumber- sumber, ketersediaan sumber, berita posting manual jadual persembahan/show, galeri gambar, mencari gambar, pengurusan kehadiran, pengurusan akaun society, galeri video, mencari video, mengurus ahli-ahli profil,permintaan pada persembahan, pengurusan persembahan, pengurusan aduan society,bantuan dan maklumat, jadual individual, pengurusan sumber/aset dan pengurusan portal. Meletakan V-model dalam iteraton and incremental development process digunakan untuk membangunkan projek ini. Ini terdapat 2 fasa iaitu fasa pengesahan dan fasa pengesahan. Fase pengesahan terdapat requirement analysis, system design architecture design dan coding. Fasa pengesahan mempunyai have unit testing, integration testing, system testing and acceptance testing. Dua puluh modul ini akan dibahagikan kepada bahagian-bahagian bebas iaitu 6 bahagian, keutamaan set ini juga diatur dalam susunan yang dalam set pertama adalah perkhidmatan keutamaan tertinggi. Sistem ini Berjaya mencapai objektif dan menyelesaikan masalah- masalah yang dinyatakan sebelum ini.

TABLE OF CONTENTS

	Page
SUPERVISOR’S DECLARATION	i
STUDENT’S DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF APPENDICES	xvi
LIST OF ABBREVIATIONS	xvii
CHAPTER 1 INTRODUCTION	
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scopes	3
1.5 Thesis Organization	6
CHAPTER 2 LITERATURE REVIEW	
2.1 Comparison of selected methodology	7
2.1.1 Spiral Model	8

2.1.2	Waterfall Model	10
2.1.3	Iterative & incremental Development	12
2.2	Justification of selected methodology	17
2.3	Survey on the existing system of 24 seasons drums society	21
	System that similar with 24 season drums society management system	22
2.3.1		
2.3.2	Malaysia dog club management system	22
	Heuristic on Malaysia dog club management system	26
2.3.3		
	Michael Jackson fan club management system	28
2.3.4		
	Heuristic on Michael Jackson fan club management system	30
2.3.5		
2.4	Development tools	31
2.4.1	Programming Language	31
	2.4.1.1 Hypertext Pre-processor (PHP)	31
	2.4.1.2 Active Server Pages.NET(ASP.NET)	32
	2.4.1.3 Java Server Pages (JSP)	32
	2.4.1.4 Comparison of advantages and disadvantages of programming language	33
2.4.2	Integrated development environment (IDE)	34
	2.4.2.1 Adobe Dreamweaver CS5	34
	2.4.2.2 Microsoft visual studio 2010	35
	2.4.2.3 Netbean IDE 6.9	35
	2.4.2.4 Comparison of advantages and disadvantages of integrated development environment (IDE)	36
2.4.3	Database	37
	2.4.3.1 MySQL	37
	2.4.3.2 Microsoft SQL Server 2008 R2	37
	2.4.3.3 Oracle Database 11i/11g	38
	2.4.3.4 Comparison of advantages and disadvantages of database	39

CHAPTER 3 METHODOLOGY

3.1	Over View to the Model	40
	3.1.1 The way to adopting V model in iterative & incremental development	40
3.2	Initial Planning	42
3.3	Verification Phases	44
	3.3.1 Requirements analysis	44
	3.3.2 System Design	47
	3.3.2.1 Entity-Relationship Diagram	48
	3.3.3 Architecture Design	53
	3.3.3.1 List of module and functionality	53
	3.3.3.2 Workflow Design	56
	3.3.3.3 Interface Design	62
	3.3.3.4 Data flow design	63
3.4	Validation	68
3.5	Software and Hardware Specification	68
	3.5.1 Software Specification	69
	3.5.2 Hardware Specification	70

CHAPTER 4 IMPLEMENTATION

4.1	Database Implementation	69
	4.1.1 Database and Server Connection	70
4.2	System implementation by using combination of V-model and iteration & incremental	70
4.3	System Interface and Implementation	72
	4.3.1 Main Page/ Login	72
	4.3.2 Member Profile – Member	73
	4.3.3 Attendance – Member	74
	4.3.4 Resources Reservation	75
	4.3.5 New and Comments	76
	4.3.6 Performance List	78

4.3.7	Individual Timetable	79
4.3.8	Pictures	80
4.3.9	Complaint	81
4.3.10	Help and Information	82
4.3.11	Reservation History	82
4.3.12	Advertisement	83
4.3.13	Search	84
4.3.14	Main Page – Administrator	85
4.3.15	Main Page Edit	85
4.3.16	Advertisement Management	86
4.3.17	Reservation Management	87
4.3.18	Profile Management	89
4.3.19	Attendance Management	90
4.3.20	Resources Management	91
4.3.21	Performance Management	94
4.3.22	Pictures Management	95
4.3.23	Complaint Management	96

CHAPTER 5 RESULT AND DISCUSSION

5.1	User Acceptance Testing	97
5.1.1	Software Tester	98
5.1.2	User Acceptance Testing Result	98
5.2	System Testing	99
5.2.1	System Tester	99
5.2.2	Testing Modules	100
5.2.2.1	User Login	100
5.2.2.2	Search Function	102
5.2.2.3	Post New and Reply	104
5.2.2.4	Resources Reservation	104
5.2.2.5	Profile Management	106
5.2.2.6	Resources Management	108
5.2.2.7	Reservation Management	109

	5.2.3 System Testing Overall Result	110
5.3	Discussion and Analysis on the Outcomes	111
5.4	Constraints	112

CHAPTER 6 CONCLUSION

REFERENCES	114
-------------------	-----

APPENDICES	162
-------------------	-----

A	Gantt Chart	118
B	Implementation (coding)	121
C	Acceptance Testing and System Testing	
D	Sign Off and Acknowledgement	142

LIST OF TABLES

TABLE NO.	Title	4 Page
2.1	Heuristic evaluation for website on Malaysia Dog Club Management System	26
2.2	Heuristic evaluation for website on Michael Jackson Fan Club Management System	29
2.3	Comparison of Advantages and Disadvantages of Programming Languages	32
2.4	Comparison of Advantages and Disadvantages of Programming Tools	34
2.5	Comparison of Advantages and Disadvantages of Database	37
3.1	Software Specification for 24 Season drums society	67
3.2	Hardware Specification for Online Industrial Sales and Support System	68
5.1	Tester Details	98
5.2	Summary of User Acceptance Testing Result	99
5.3	Tester Details	100

LIST OF FIGURES

FIGURE NO.	TITLE	PAE
2.1	Spiral Model	9
2.2	Waterfall Model (Ain Sommerville, 2001a)	11
2.3	Iterative development models (Wikipedia, 2011a)	13
2.4	V model	14
2.6	combination of V-model and iteration & incremental	18
2.7	Measuring increments in successfully tested software (IBM developer Works, 2011)	19
2.8	Malaysia dogs club management System - Main Page	22
2.9	Malaysia dogs club management System – Help / Reference	23
2.10	Malaysia dogs club management System – Articles store	24
2.11	Michael Jackson Fan Club Management System main page	27
2.12	Michael Jackson Fan Club Management System latest new page	28
3.1	Combination of V-model and iteration & incremental	39
3.2	Entity-Relationship diagram (ERD) for whole system	47
3.3	Entity Diagrams for first incremental set	48
3.4	Entity Diagrams for second incremental set	48
3.5	Entity Diagrams for third incremental set	49
3.6	Entity Diagrams for fourth incremental set	49
3.7	Entity Diagrams for fifth incremental set	50
3.8	flow chart of 24 season drums society for existing member, future member and open user	55
3.9	flow chart for first incremental set	56
3.10	flow chart for second incremental set	57
3.11	flow chart for third incremental set	58

3.12	flow chart for fourth incremental set	59
3.13	flow chart for fifth incremental set	60
3.14	User Interface Flow Diagram for 24 Season drums society	61
3.15	Context Diagram of 24 Season Drums Society Management System	62
3.16	DFD Level 0 of 24 Season Drums Society Management System for existing member	63
3.17	DFD Level 0 of 24 Season Drums Society Management System for open user	64
3.18	DFD Level 0 of 24 Season Drums Society Management System for administrator	65
4.1	24SDSMS Database	70
4.2	24SDSMS Main Page	72
4.3	24SDSMS Member profile – member	73
4.4	24SDSMS Edit Member profile – member	74
4.5	24SDSMS Attendance List – Member	74
4.6	24SDSMS Resources Reservation – Member	75
4.7	24SDSMS Resources Reservation – Member	75
4.8	24SDSMS Resources Reserve confirmation – Member	76
4.9	24SDSMS New and Comment – Member	77
4.10	24SDSMS New and Comment1 – Member	77
4.11	24SDSMS New and Comment2 – Member	78
4.12	24SDSMS Performance Timetable– Member	78
4.13	24SDSMS Performers Name List– Member	79
4.14	24SDSMS Individual Timetable– Member	79
4.15	24SDSMS Performers Name List – Member	80
4.16	24SDSMS Albums– Member	80
4.17	24SDSMS Pictures– Member	81
4.18	24SDSMS complaint – Member	81
4.19	24SDSMS help and information – Member	82
4.20	24SDSMS Reserved History– Member	83
4.21	24SDSMS Advertisement List– Member	83
4.22	24SDSMS Advertisement Content – Member	84

4.23	24SDSMS Search Function – Member	84
4.24	24SDSMS Main Page –Administrator	85
4.25	24SDSMS Main Page Edit –Administrator	86
4.26	24SDSMS Advertisement Manager –Administrator	87
4.27	24SDSMS Reservation Manager –Administrator	88
4.28	24SDSMS Reservation Manager –Administrator	88
4.29	24SDSMS Reservation Manager –Administrator	88
4.30	24SDSMS Member Profile Manager –Administrator	89
4.31	24SDSMS Member Profile Manager –Administrator	90
4.32	24SDSMS Attendance Management –Administrator	91
4.33	24SDSMS Attendance Management –Administrator	91
4.34	24SDSMS Resources Management –Administrator	92
4.35	24SDSMS Collect Resources –Administrator	92
4.36	24SDSMS Return Resources –Administrator	92
4.37	24SDSMS Reservation History –Administrator	93
4.38	24SDSMS Collect Resources Receipt –Administrator	93
4.39	24SDSMS Return Resources Receipt –Administrator	93
4.40	24SDSMS Performance Management–Administrator	94
4.41	24SDSMS add Performers–Administrator	95
4.42	24SDSMS Create album–Administrator	95
4.43	24SDSMS Add Picture–Administrator	96
4.44	24SDSMS Complaint Management –Administrator	96
5.1	Testing Result for user login function	101
5.2	Testing Result for user login function	103
5.3	Testing Result for Post new and reply	104
5.4	Testing Result for Resources Reservation	105
5.5	Testing Result for Profile Management	107
5.6	Testing Result for Resources Management	108
5.7	Testing Result for Reservation Management	109
5.8	Summary of the result of Oracle Application Testing Suits	110
5.9	Summary of the result of Oracle Application Testing Suits Admin	111

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Gantt Chart	118
B	Implementation (coding)	121
C	Acceptance Testing and System Testing	142
D	Sign Off and Acknowledgement	147

LIST OF ABBREVIATIONS

PHP	Hypertext Preprocessor
ASP	Active Server Pages
JSP	Java Server Pages
IDE	Integrated Development Environment
SQL	Standard Query Language
HTTP	Hypertext Markup Language
SDD	Software Design Document
DFD	Data Flow Diagram
ERD	Entity-Relationship Diagram

CHAPTER 1

INTRODUCTION

This chapter briefly discuss on the overview of this research. It contains five sections. This first section is introduction; follow by the problem statement. Next are the objectives where the project's goal is determined. After that are the scopes of the system and lastly is the thesis organization which describes the structure of this thesis.

1.1 Background

Today's technologies progressing rapidly than everyone can imagine and keep changing our life style, especially internet service. With the rapid evolution of Information Technologies (IT), especially Internet & Intranet. To many enterprises, the importance of the IT has been unceasingly increasing, more and more getting into the main business of an enterprise in depth. IT eventually becomes a vital part of the enterprise's core competence, which will definitely have a great influence on the decision making and the development strategies of an enterprise in a long run.(jiang Hongxun,2006) The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. And these networks consists of millions of private, public, academic, business, management and government networks. There is a huge amount of information available on the internet for just about every subject known to man, ranging from government law and services, trade fairs and conferences, market information, new ideas and technical support.

Apply the management system into online system and let technologies to replace paper waste also an another usage of internet. Because of the internet technology, more and more companies prefer to apply their management system through online system. Because building customer relationships is of critical strategic importance to firms in achieving success in the marketplace (Ryals, 2005). The usage of internet is so wide and many business companies try to make full use of it services when dealing with customers. Through web management system, users feel that they not only can save their time and cost, at the same time they can get any information that they want. Because of these features, other than companies, many organizations and societies also follow the step in this rapidly progress technologies world.

Therefore, the 24 season drums society management system is developed to make flow of management can process with more efficiency. The system will provide basic information and background of the society. And the system is separated to 3 different types of users such as open user, member and admin. The system is strictly controlled by the admin.

1.2 Problem Statements

Society members are increasing rapidly due to the increment on the students enrollment. However, society management mostly being done by manually.

It means that all data in each progress all using paper to record so the costs to acquire the society are relatively expensive.

1.3 Objectives

The objectives of this project are:

- To developing the 24season drums society management system using iterative & incremental approach.
- To automate the managing process of 24 season drums society.

1.4 Scopes

The scopes of this project are:

- i. Manage donation
 - to provide a site to let open users or members to get more information about donation
 - Admin has to manage the donator information such as their personal profile information.
 - Admin has to update account balance and make a confirmation once transaction was successful.
 - Provide a site to let public user to give a donation or sponsor.

- ii. resource booking
 - To provide a form fill-in site to let 24 season drums society members apply event/tools.
 - To provide a form fill-in site to let members request for venue booking for training or practice.

- iii. Resource availability
 - Check for latest condition of tools and record.
 - Admin has to update for quantity and quality of tools.

- iv. Posting news and events
 - To provide a site to let admin or members to post the latest information and events

- v. Add comments to news & events
 - To provide a post comment space for 24 season drums society members to post their comments to share their opinion.

- vi. Manual performance/show timetable
 - Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the performance and show information.
 - Provide a calendar that full of event schedule to let members and open users to view all the upcoming events.

- vii. Picture gallery
 - Admin able to update the Picture gallery by category to several album.
 - Members and open users can view the photos that snap from performance and others activity.
 - Admin able to add in the information of the photo such at time, performance venue and name of members in photo

- viii. Search on photo
 - Provide search function to let users search wanted photo by key in the related information such as members name, venue and name of performance.

- ix. Attendance management
 - To record the attendance for each training or activity.

- x. Management for society account
 - To manage the member fees and other spent cost.
 - To record all the spent cost and income fees.

- xi. Video gallery
 - Admin able to update the video gallery by category to several album.
 - Members and open users can view the video that recorded from performance and others activity.
 - Admin able to add in the information of the video such at time, performance venue and name of members in photo.

xii. Search on video

- Provide search function to let users search wanted video by key in the related information such as members name, venue and name of performance.

xiii. Manage members profile

- Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the member profile.
- Members will provide an ID and password to log in to the website.
- Members can update their profile manually.

xiv. Request for show and performance

- Open user can request for show and performance by key in venue and time to waiting for approval.
- Admin has to check for availability for performance date and discuss with group members by posting new.
- Admin has to reject or approve the request once have an answer for discussion.

xv. Performance management

- Admin has to list down all preparation for each upcoming performance such as uniform and tools need to use.
- Manage for transportation when needed.

xvi. Society complaint management

- To provide a site to let users to make a complaint for anything that related to society to improve the efficiency.

xvii. Help and information

- Provide help and information to solve user problem when using this website.
- Introduce about 24 season drum society to let all user more understand the purpose of 24 drums performance.

xviii. Self timetable

- Let members can manage their own schedule by themselves such as view their own involved performance date and time.

xix. Resource/asset management

- Allow administrator to manage to resources

xx. Portal management

- Allow administrator to manage the portal.

1.1 Thesis Organization

This thesis consists of four (4) chapters. Chapter 1: Introduction briefly describes and introduces the system. This system preliminary shows the basic concept of the system, problem statements of the system, objectives, scopes, and how the report is organized. Chapter 2: Literature Review depicts the manual systems and the existing systems as the case studies of the project. This chapter also reviews the technique, method, equipment, and technology that had been used in the case studies. Chapter 3: Methodology discusses about the overall workflow in the development of the project. This chapter also discusses the method, technique or approach that has been used while designing and implementing the project. Chapter 4: Conclusion briefly summarizes the project.

CHAPTER 2

LITERATURE REVIEW

This chapter will be devoted to a survey from selected methodology and way to adopting, followed by the existing systems that similar with 24 Seasons drums society management system and Support System, development tools.

2.1 Comparison of Selected Methodology

Methodology is an ongoing process where software developers used as guidelines to build system from goal definition to the actual system or final product. It is important to identify and use a suitable methodology that suits the development of the project the developer is doing to ensure every phase and stages are rightly focus and apply to achieve project goals set. Three software methodologies had been identified and considered for the development of Online Industrial Sales and Support System which is as below:

- i) Spiral Model
- ii) Waterfall Model
- iii) Iterative & Incremental
- iv) V Model

2.1.1 Spiral Model

Spiral model was originally created by Barry Boehm in his 1988 article A Spiral Model of Software Development and Enhancement to address the inadequacies of the Waterfall Model and it is also referred as Boehm-Spiral software engineering methodology. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive, and complicated projects. The essential and major concept of Spiral model is to reduce risk by the frequent usage of prototypes. The spiral model works and starts at the centre of the spiral.

According to Boehm, “the major distinguishing feature of the Spiral Model is that it creates a risk-driven approach to the software process rather than a primarily document-driven or code-driven process. It incorporates many of the strengths of other models and resolves many of their difficulties” (Boehm, 1988). As shown in Figure 2.1 (Online Interactive Modules for Teaching Computer Science, 1997), each completed cycle along the spiral represents one stage of the process. With each loop of the spiral, the customer can evaluate the work done to the project so that the customer can present suggestions for modification to be done. As the spiral process continues, the software is further developed and enhances to make it more mature and in line with the project goals and requirements.

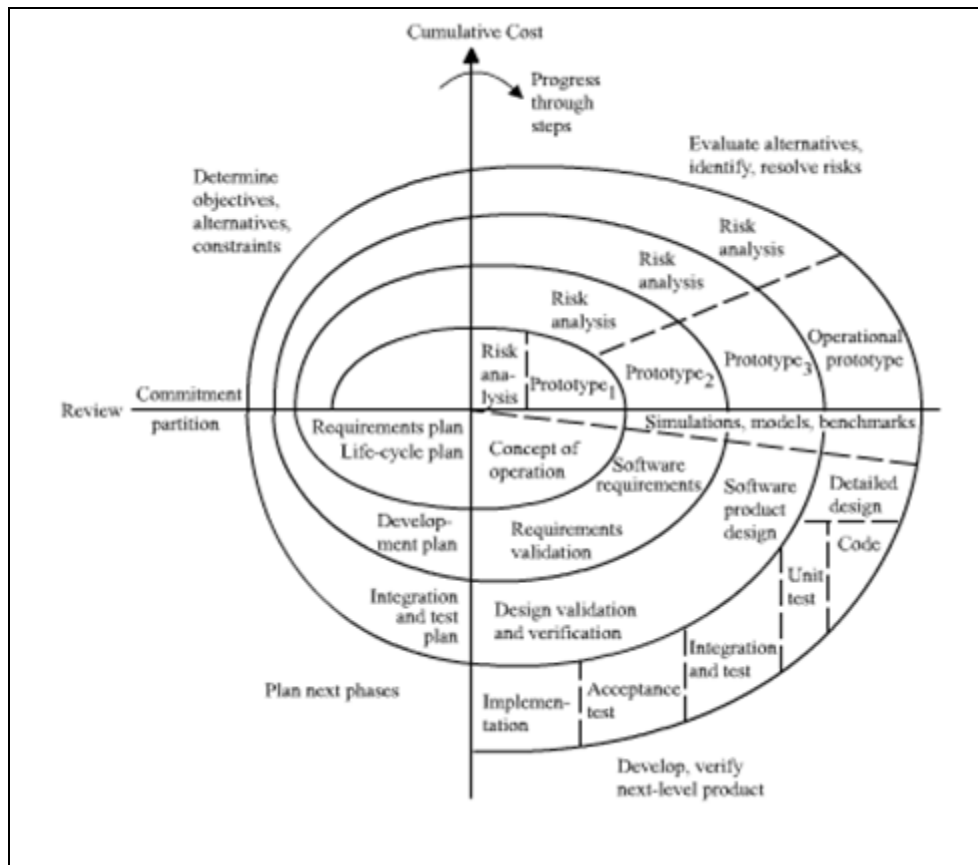


Figure 2.1: Spiral Model

The steps of Spiral Methodology (Freetutes, 2007):


- i) System requirements are defined in the most detail ways as possible which usually involves interviewing a number of users that are vital in the aspects of the existing system.
- ii) A preliminary design is created for the new system.
- iii) Prototype of the system is constructed from the preliminary design and usually is a scaled-down system and represents an approximation of the characteristics of the final system.
- iv) Second prototype is released and evolved based on four steps which are, evaluating the first prototype and identify its strengths, weaknesses and risks. Define the requirements of the second prototype. Plan and design the second prototype and finally construct and test the second prototype.

- v) Project might be aborted when the risks identified are too great because risk factors could cause development cost to overrun, miscalculation of operating cost and will end up providing a final product that fails to meet its goals and requirement set in the beginning of the project.
- vi) Existing prototype will be evaluated just like how the prototype was evaluated earlier in the steps and if necessary another prototype will be developed.
- vii) Steps are iterated until the customer or user is satisfied that the refined prototype represents the final product desired and then the final system will be constructed based on the refined prototype.
- viii) Finally, the final system will be thoroughly evaluated and tested and routine maintenance is carried out to identify any problems that surfaces and rectify it to enhance the system and make it more stable and in the same time minimizing serious failures and downtime.

2.1.2 Waterfall Model

The waterfall model is a first model of the software development process was derived from other engineering process (Royce, 1970). It is a model which was developed for software development that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall.

Waterfall model has been structured on various phases especially to help out the software construction companies to develop an organized system of construction. The project will divide into many stages by following this method. When start with first Phase and according to this model, only can proceed to next phase once the previous one has been completed. This way one moves progressively to the final stage and once that point is reached, then cannot turn back, similar to the water in a waterfall.



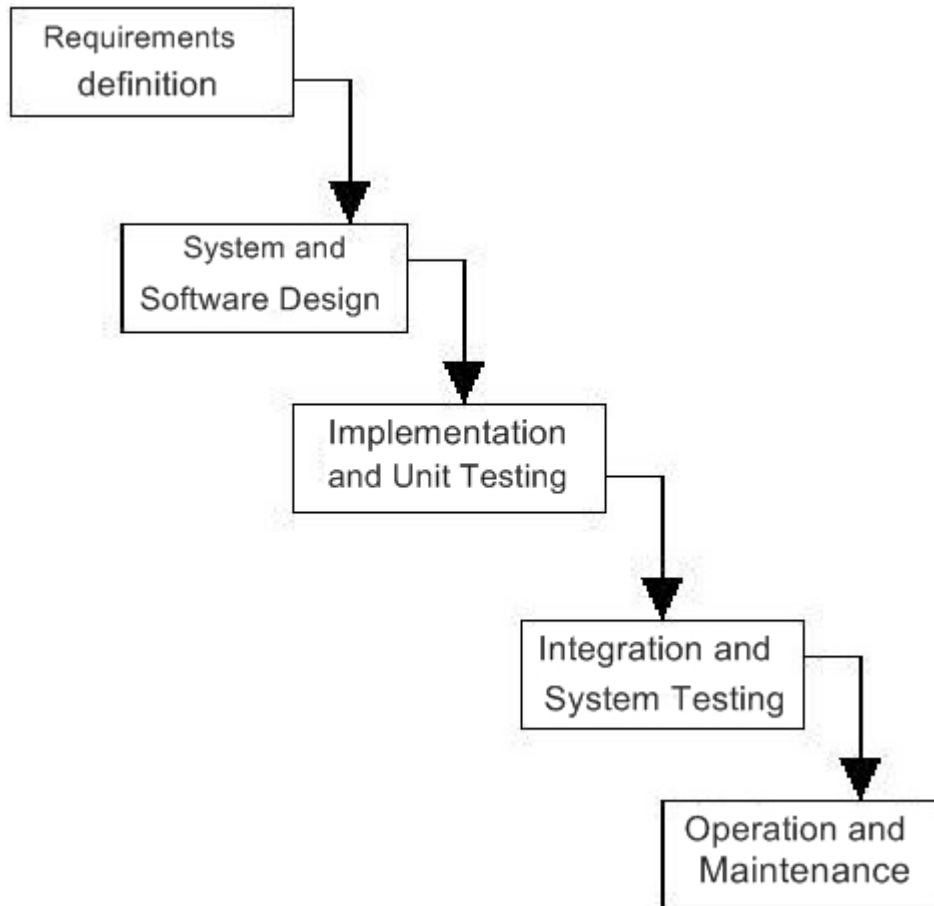


Figure 2.2: Waterfall Model (Ain Sommerville, 2001a)

According to Figure 2.2, waterfall model takes the fundamental process activities of specification, development, validation and evolution and represents them as separate process phase such as (Ain Sommerville, 2001a):

i. Requirements Analysis and Definition

The system's services, constraints and goals are established by consultation with system users. They are defined in detail and serve as a system specification.

ii. System and Software Design

The systems design process partitions the requirements to either hardware or software system. It establishes an overall system architecture. Software design

involves identifying and describing the fundamental software system abstractions and their relationships.

iii. Implementation and Unit Testing

During this stage, the software design is realised as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

iv. Integration and System Testing

The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

v. Operation and Maintenance

Normally this is the longest life cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's services as new requirements are discovered.

2.1.3 Iterative & Incremental Development

Iterative and Incremental development is developed in response to the weaknesses of the waterfall model in which at heart of a cyclic software development process. It starts with an initial planning and ends with deployment with the cyclic interactions in between. The incremental approach to development was suggested by Mills (Mills *et al.*, 1980) as a means of reducing rework in the development process and giving customers some opportunities to delay decisions on their detailed requirements until they had some experience with the system.

A basic idea of Iterative & Incremental Development is develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental),

the purpose of this development is allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

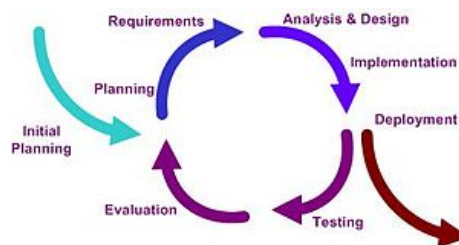


Figure 2.3

Figure 2.3: Iterative development models (Wikipedia, 2011a)

Incremental development process has a number of advantages:

- Customers do not have to wait until the entire system is delivered until they can gain value from it. The first increment satisfies their most critical requirements so the software can be immediately used.
- Customers can use the early increments as a form of prototype and gain experience which informs the requirements for later system increments.
- There is a lower risk of overall project failure. Although problems may be encountered in some increments, it is likely that some will be successfully delivered to the customers.

However, there are some problems with incremental development. Increments should be relatively small (no more than 20,000 lines of code) and each increment

should deliver some system functionality. It may therefore be difficult to map the customer's requirements onto increments of the right size.

2.1.4 V-Mode

The V model is modified from waterfall method. As opposed to the Waterfall method, the design of this model is not in a linear axis and is the stages turn back upwards after the coding phase is done and the shape is similar with V shape hence the name V model is used. It was put forth by Paul E. Brook in 1986. And it have different stages. This developmental process is balanced and relies on the verification from the previous steps before proceeding forwards. When the product from one phase has reached completion, it will then form the basis for the next phase. Similar to the waterfall model, one progresses to the next step when the previous one has been completed. But in the V model, we need to check and approve the product from every phase before moving forward.

The biggest advantage of using the V Model is that unlike the Waterfall method and the aorta life cycle method, every stage is tested.

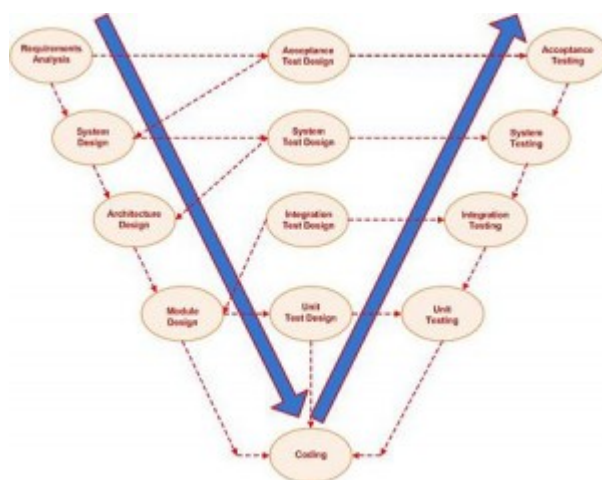


Figure 2.4:V model

According to figure 2.4 V model, there are various Stages of the V model which are (V model,2011):

Verification Phases

- i. Requirments analysis
 - This is the first step in the verification process. It is in here that decide the project and its function.
 - During this stage the employees are not going to discuss how it is going to be built but discussion and a user requiremnet is needed. The function of the system, performance, security, data, interface and needed in this document.

- ii. System Design
 - System design is the phase whre system engineers analyze and understand the propeties of the sytem by studying the user requirements document.
 - The software specification document which serves as a blueprint for development phase is generated. This document contains the general system organization, menu structures, data structures.

- iii. Architeture Design
 - Consists of the list of modules, brief functionality of each module, interface relationships, dependencies, database tables, arhitecture diagrams.

- iv. Module Design
 - The low level design document or program specifications will contain a detailed functional logic of the module, in pseudocode.
 - Database tables, all interface details with complete API refreences, all dependency issues, error message listings and complete input and ouputs for a module also needed in this stage.

Validation Phases

- i. Unit Testing
 - Unit tests are created by programmers or occasionally by white box testers.
 - The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage.

- ii. Intergration Testing
 - In intergration testing the separate modules will be tested together to expose faults in the interfaces and in the intraction between intergrated components.
 - Testing is usually black box as the code is not directly checked for errors.

- iii. System testing
 - System testing will compare the system specifications against the actual system.

2.2 Justification of Selected Methodology

After an in depth research conducted on Waterfall Model, Spiral Model, Iterative & Incremental and V Model, the chosen methodology for this particular project is adopting V model in Iterative & Incremental development.

Iterative & Incremental development is heart of a cyclic software development process developed in response to the weaknesses of the waterfall model. Reflection or revision are does not allowed much in Waterfall development. The problem is its inflexible partitioning of the project into these distinct stages. Commitments must be made at an early stage in the process and this means that it is difficult to respond to changing customer requirements. Once an application is in the testing stage, it is very difficult to go back and change something that was out of the concept stage. Therefore, the waterfall model should only be used when the requirements are well understood.

However, the advantages of the waterfall model are that it is a simple management model and its separation of design and implementation should lead to robust systems which are amenable to change. By contrast, an evolutionary approach to development allows requirements and design decisions to be delayed but also leads to software which may be poorly structure and difficult to understand and maintain.

In contrast, Incremental and iterative development process exploits the fact that a big project is breaking into smaller pieces and we can start a simple build and work around in initial set to gaining experience and knowledge, and iteration is applied to help us to learning advantage from build to build .The system design and implementation work must be reworked to implement the changed requirement. In each iteration of development process, design modifications are made and new functional capabilities are added.

The advantage of V-model is that it saves much amount of time and since the testing stage is involved early on, thus can develop a very good understanding of the project at the very beginning when compare to using Spiral Model which is more towards large and complex projects.

But at each stage of V – model there is a potential of errors. The first testing is done after the design of modules which is very late and costs a lot. So break the project to various parts by incremental development and these all parts will work simultaneously. By break the project to small pieces also can make the testing more easily due to the range for tester to test the system become smaller. Using incremental development into V model not only save lot of time, and it also can let us learning advantage from build to build can improve the weakness from previous stage. Besides that, for iteration progress can allow developer to do any changes and add new functional capabilities according to customer requirement once any mistakes occur.

2.2.1 The way to adopting V model in Iterative & Incremental development

Let consider incremental development at first, the project is divided to various independent parts by incremental development and the division is as much as possible. Assuming a example of my development of a society management website with parts like member registration, member login, view profile, view video, post comments and resource booking. Depending on the product owners requirements, I can start with member registration, member login and view profile by complete these and during this stage, adopting V- model inside incremental development, each phase in V-model is need to complete. Once these part are ready, then can proceed to the next set. These all parts are also possible to working simultaneously and integrating them when ready in the central repository.

The combination of v-model and incremental development process would be like this:

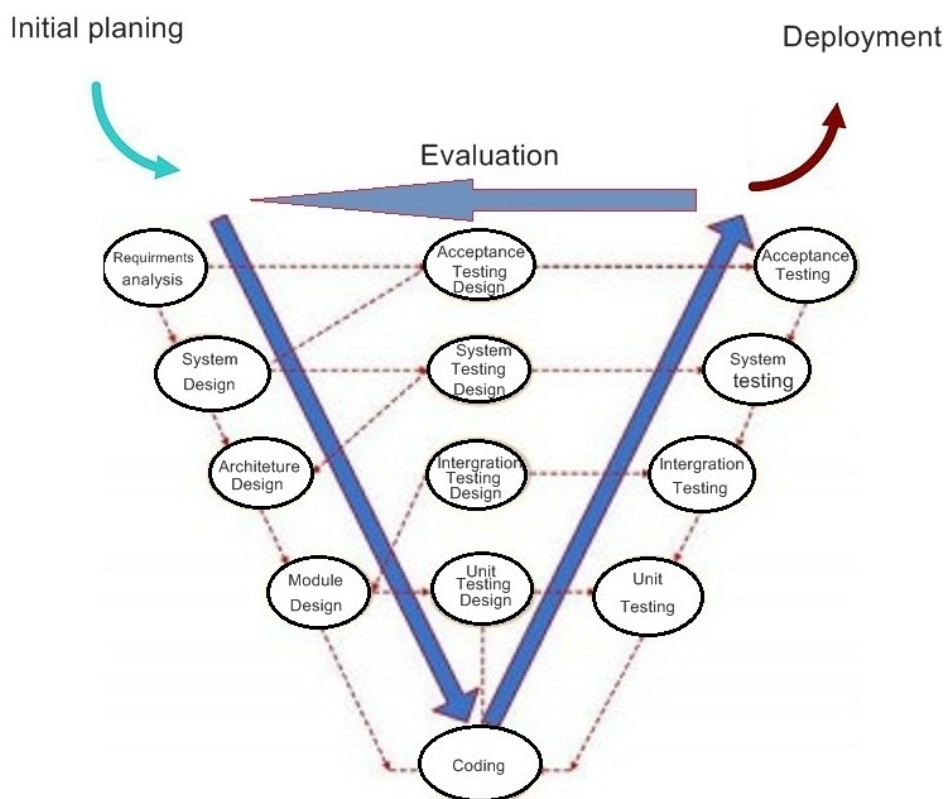


Figure 2.6: combination of V-model and iteration & incremental

From figure 2.6, the combination of V- model is not much different with previous original V –model, there are only a little changes such as initial planning , deployment when testing are done and evaluation for next iteration. In incremental process, developer needs to identify which of the services are most important and which are least important to them.

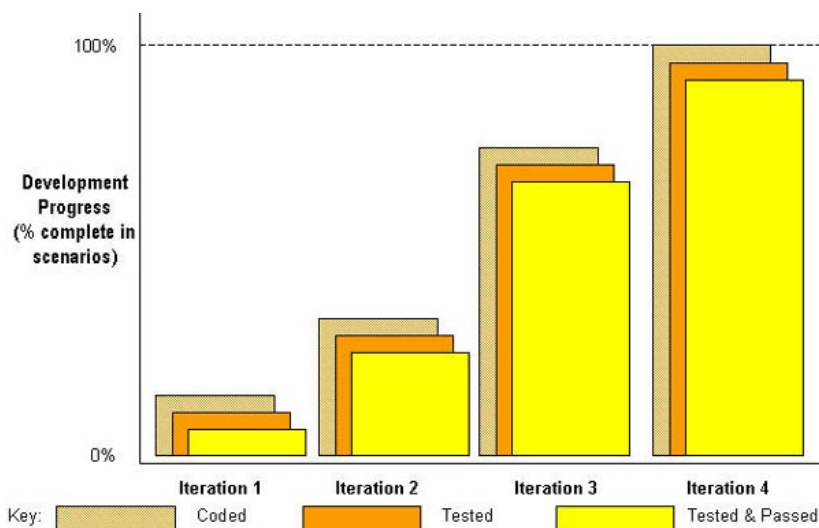


Figure 2.7: Measuring increments in successfully tested software (IBM developer Works, 2011)

According to data from figure 2.7(IBM developer Works, 2011), the progress are done by various iteration and progress which done for each iteration are getting more until 100% complete for one incremental set.

By using the same example, one way of looking at iterative development is to see the first iteration comprising of member registration, member login and view profile. The second iteration can comprise of photos, photos search, video, and video search. And a better way to look iteration would be like this:

- Provide member registration with form fields like name, email address, password and submit button.
- In the next “iteration” can divide name into first name and last name, email address and password to be confirmed twice and eliminate zip code entirely and add birth date to the registration.

2.3 Survey on the Existing System of 24seasons Drums Society

24season drums society has an existing system to manage all the member profile and all data of society .The society has a manual system that only provide some basic information of member, records of activities of the company and some important contact number. They do not utilize the usage of internet services in their society and the system did not provide any extra services for members and open user that seeking the information. As a result, the members and open user are difficult to taking the latest information of society.

The society is using personal detail form to record down the members' information manually and keeps them in a file. They do not have an organized member's database and it is hard to manage and search back the members by using filing method. Sometimes the information of the members may lose because of some unexpected accident. The society is planning to register all members in computer and store them safely.

The society has a clerk responsible for picking up people that interest in request performance's phone call for appointment booking and enquires for information. The clerk record down all the appointment manually on a note and pass to the secretary and president. It is a very traditional way to record down appointment and it may cause misunderstanding between the clerk and people that interested because it does not have a proper way of recording down the appointment and sometimes conversation can brought to conflict. When passing the note to the secretary, they may misplace the note or the time and date for the appointment crash with other appointment.

The members or president of the company will deal with the people who interested face to face regarding on the performance customization. Once again, the requirements are wrote down by the secretary manually on a paper and brought back to the society to plan for performance requirement that set by both party and after they finish up, they set a date and time to deal again for confirmation. Eventually these processes need to repeat a few times to discuss with the people who interested to make sure the requirements are met. It is wasting time and cost for the society because the requirements cannot finalize early and do not have a clear overview plan.

2.3.1 Systems that Similar with 24 Season Drums Society Management System

There are a lot of sales and support system online and the best three that are similar with the Online 24season drums society management system function were chosen to do critique and comparison.

2.3.2 Malaysia Dog Club Management System

The Malaysia dog club management system is a website that mainly interacts with all dogs' lovers by posting comments, blogs, video, pictures and others.

The Malaysia dog club management system main page as shown in Figure 2.8 (Malaysia Dog Club, 2011a) consists of navigation bar that link to other function of the system. It provide login and account registration function on the page. The page is customizable for the existing user to choose what content to display on the main page. It also provides account and profile setting for existing user. Call is available for user to contact their company instantly by clicking the link at the page.

The main page eases the existing user to login and customize their preferable page, account and profile setting. The navigation bar is well organized to increase the usability of the system. But the system lack of brief introduction of the club and

instruction to guide them at the main and eventually will confuse the user which is first time visiting.

The main page also showing the recent activities of member who post out the comments and videos. Search tools are prepared for user to search specific information. The numbers of members, category and active group also shown on sidebar.

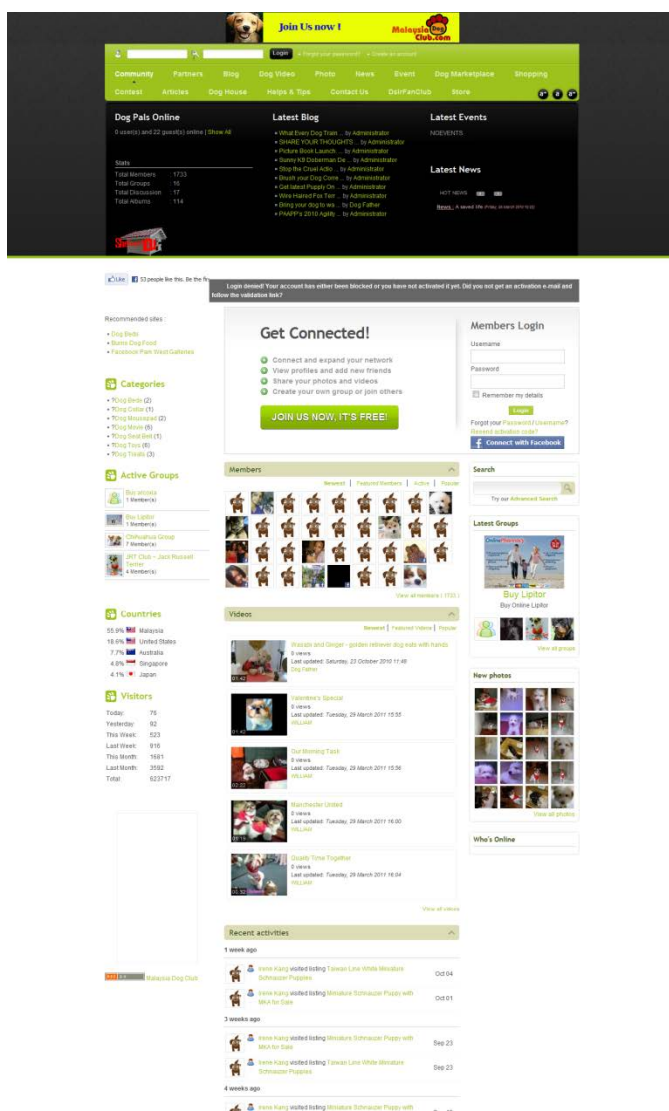


Figure 2.8: Malaysia dogs club management System - Main Page



Figure 2.9: Malaysia dogs club management System – Help / Reference

The Malaysia Dogs Club Management System Help / Reference page as shown in Figure 2.9 (Malaysia Dog Club, 2011a) consists lot of reference and info such as terms and conditions for Malaysia Dog Club's photo contest, but the contest are launch in 2009, it means that admin no longer update their reference and help page since 2009. Here also shown for instruction for user that wish to upload photos. Instruction for edit profile info and how set cover photo for photo albums also provided.

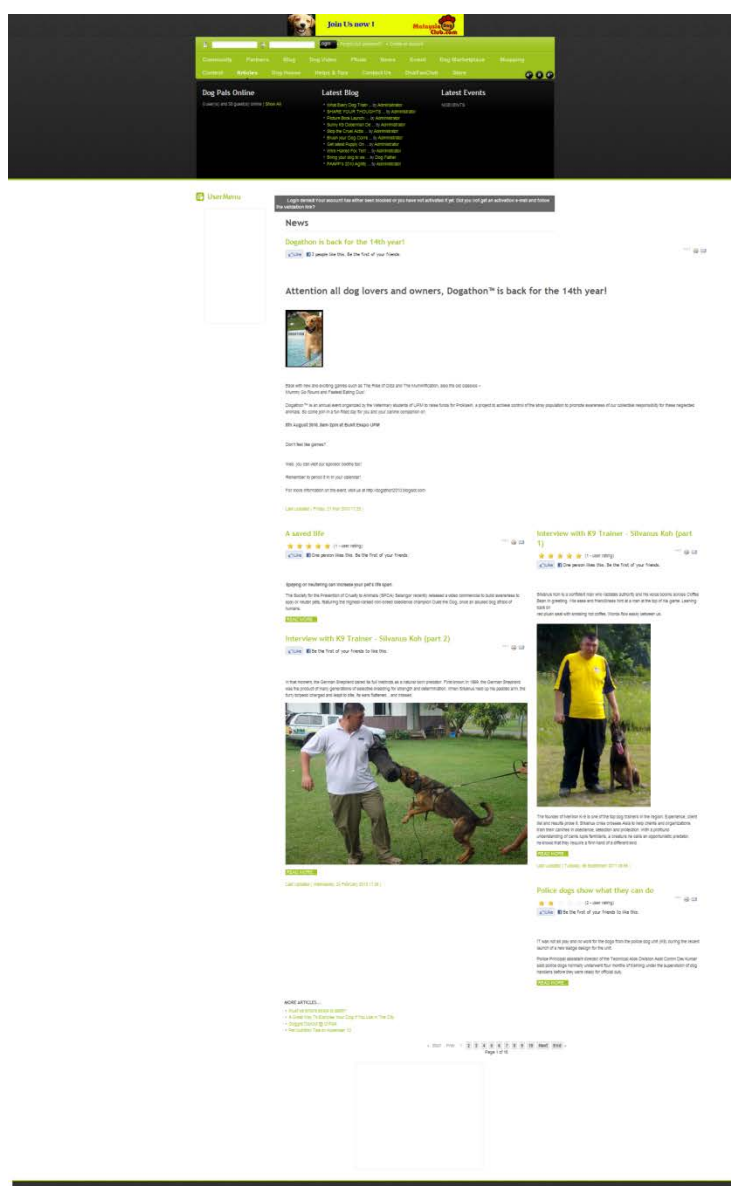


Figure 2.10: Malaysia dogs club management System – Articles store

Malaysia Dogs Club Management System articles site as shown in Figure 2.10 (Malaysia Dog Club, 2011a) consists of various type of solution article for the pet owners to view and study. Those articles are gather from all over the world wrote by experts.

There is a large database full of articles all around the world for the user to search for dogs' information. And it provides direct access to view those solutions free. But it is quite hard and waste of time to read the whole article just to get a step of solution at the end from the article. It does not provide simple step by step guidance for some common solutions.

2.3.3 Heuristic on Malaysia Dog Club Management System

According to these three web pages from Malaysia dogs club management System; some of the user-interface elements are not conforms to the principles. User interface can include dialog boxes, menus, navigation structure, online help, and so on. For example, from The Malaysia dog club management system main page as shown in Figure 2.8(Malaysia Dog Club, 2011a) . Even though navigation bar are used for user to increase the usability of the system. But the system lack of brief introduction of the club and instruction to guide them at the main and eventually will confuse the user which is first time visiting and it is not conform the “support internal locus of control” in Shneiderman’s 8 Golden rules (Shneiderman, 1998) due to inability to obtain or difficulty in obtaining necessary information. The arrangement of data and information to show on first page also too complicated. For example latest blog, latest even, latest news and number of members online are cover for whole display window. For the main information that suppose to show out become hardly to figure out due to users are need to scroll down for further information and the information to display is more enough until users have to scroll more than 4 times for one page. And it is not conform the “reduce short-term

memory load” rule that stated in Shneiderman’s 8 Golden rules (Shneiderman, 1998) because the limitation of human information processing in short-term memory requires to displays be kept simple, multiple-page displays be consolidated, window-motion frequency be reduced.

For those 3 pages that shown, according to Norman’s 7 Principles (Norman Donald, 1998), principles that designer didn’t follow which is “simply the structure of task”. Task need to simple in order to avoid complex problem solving and excessive memory load. but for the figure 2.9(Malaysia Dog Club, 2011a) the ordering of each task are messy especially for the term & condition of contest should be remove form list because there are not categories on that category plus date for that contest are expired more than 1 year. Table 2.1 shown the heuristic that have apply in website and else.

Table 2.1: Heuristic evaluation for website on Malaysia Dog Club Management System

Strive for consistency	Yes
Simplifying the structure of task	No
Use both knowledge in the world and knowledge in the head	Yes
Reduce short-term memory load	No
Design for error	No
Offer informative feedback	Yes
Support internal locus of control	No

2.3.4 Michael Jackson Fan Club Management System

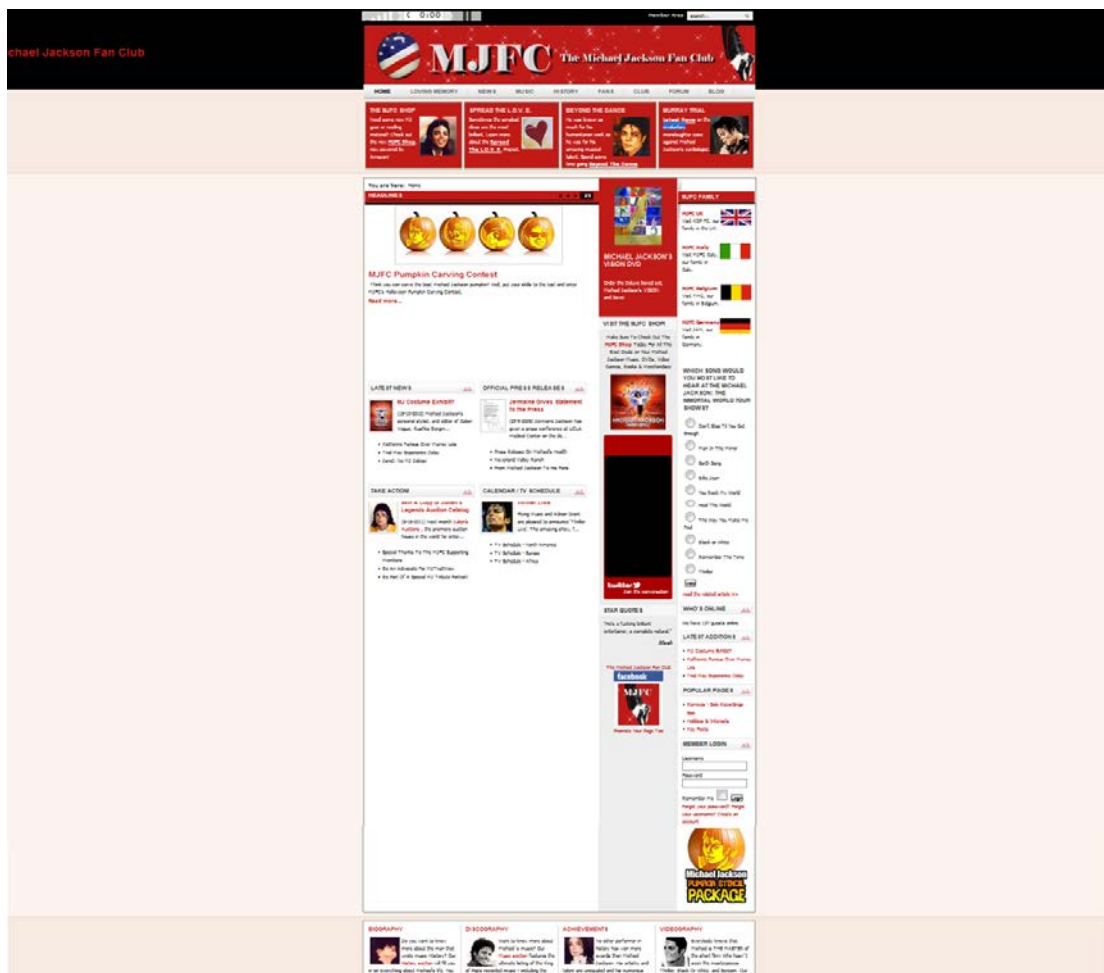


Figure 2.11 Michael Jackson Fan Club Management System main page

The Michael Jackson Fan Club Management System is a website that mainly provides a site to interact with all fans around the world that admire Michael Jackson. The latest, most reliable news on Michael Jackson (MJ) from sources worldwide also can find from here.

The Michael Jackson Fan Club Management System main page as shown in Figure 2.11 (Michael Jackson Fan Club, 2011a) consists of navigation bar that link to other function of the system. The page is providing player for playing MJ

2.3.5 Heuristic on Michael Jackson Fan Club Management System

According to these two web pages from Michael Jackson Fan Club Management System, almost all of the user-interface elements are conform to the principles. User interface can include dialog boxes, menus, navigation structure, online help, and so on. Consistent sequences of actions should be required in similar situation; identical terminology should be used in prompts, menus, and help screens; consistent colour, layout, capitalization, fonts, and so on should be employed throughout and these two web pages are follow this rule which is “strives for consistency”. Filter function and display method selection in figure 2.12 (Michael Jackson Fan Club, 2011c) make perceived system quality improve with adding features for expert. The simple layout and tidy arrangement of list of articles produce a simple display, consolidated multi-page display and window motion frequency also reduced.

Table 2.2: Heuristic evaluation for website on Michael Jackson Fan Club Management System

Strive for consistency	Yes
Simplifying the structure of task	Yes
Use both knowledge in the world and knowledge in the head	Yes
Reduce short-term memory load	Yes
Design for error	Yes
Offer informative feedback	Yes
Support internal locus of control	Yes

2.4 Development Tools

The development tools to develop the 24seasons drums society management system will be identified and choose based on which would be more suitable for the development of the project at the end of the research. And the programming language, programming tools, database and websider are in the consideration.

2.4.1 Programming Language

A programming language is an artificial language designed to express computations that can be performed by a machine (Wikipedia, 2011a).

2.4.1.1 Hypertext Pre-processor (PHP)

PHP was originally created by Rasmus Lerdorf in 1995 and since then, the language had been widely accepted in web building the development of the language has grown continuously. PHP started as a series of perl scripts for homepages and that is where its original name came from which is "Personal Home Pages". Due to its growth and to be a fully developed and accepted language and with plenty of advantages and flexibility it has been renamed to "Hypertext Pre-processor" (Jonathan Arkell, 2010).

PHP is a scripting language based on the model of pre-processing HTML pages. The PHP pre-processor in the web server will detect any PHP language tag any the PHP engine will execute that particular code. In other words, PHP is mainly focussed on server side scripting. PHP scripts are normally use in these 3 main areas which are server side scripting, command line scripting and writing desktop applications (Php.net, 2001).

2.4.1.2 Active Server Pages.NET (ASP.NET)

ASP.NET is a Web application framework that developed and marketed by Microsoft to allow programmers to build dynamic Web sites, Web application and Web services (Wikipedia, 2011). Version 1.0 of the framework was released in January 2002 and it is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET takes an object-oriented programming approach to Web page execution. According to Webopedia(2011), every element in an ASP.NET page is treated as an object and run on the server. An ASP.NET page gets compiled into an intermediate language by a .NET Common Language Runtime-compliant compiler. Then a JIT compiler turns the intermediate code to native machine code, and that machine code is eventually run on the processor. Pages load much faster than classic ASP pages due to the code is run straight from the processor.ASP.NET is used to create Web pages and Web services and is an integral part of Microsoft's .NET vision.

2.4.1.3 Java Server Pages (JSP)

Web developer creates a JSP page that includes JSP technology-specific and custom tags. JSP technology uses XML – like tags that encapsulate the logic that run the content for the page. Display, design and supporting a reusable component-based design by separating the page logic. Web-base applications can build with rapidly and easier by using JSP technology.

Java Servlet is an extension of JavaServer Pages technology. Servlet are used to fit seamlessly into a Web server framework which is a platform- independent, server-side module in purpose to extend the capabilities of a Web server with minimal overhead, support and maintenance. (Oracle, 2011a).

2.4.1.4 Comparison of Advantages and Disadvantages of Programming Languages

The table below as shown in Table 2.3 consists of comparison of advantages and disadvantages of programming language between PHP, ASP.NET and JSP.

Table 2.3: Comparison of Advantages and Disadvantages of Programming Languages

	PHP	ASP.NET	JSP
Code price	Free	Free	Free
Platform price	Free	Not free	Free
Language code style	C	VB	Java
Efficiency	Strong	Weak	Weak
Security	Strong	Strong	Weak
Platform	Any	Microsoft platforms only	Any
Source available	Yes	No	Yes
Object Oriented Programming	Yes	Yes	Yes

2.4.2 Integrated Development Environment (IDE)

An integrated development environment (IDE) also known as integrated design environment or integrated debugging environment is a software application that provides comprehensive facilities to computer programmers for software development (Wikipedia, 2011b).

2.4.2.1 Adobe Dreamweaver CS5

Adobe Dreamweaver formerly called Macromedia Dreamweaver is a web development application originally created by Macromedia, and is now developed by Adobe Systems, which acquired Macromedia in 2005. It supports various types of languages such as ActionScript, Active Server Pages (ASP), C#, Cascading Style Sheets (CSS), ColdFusion, EDML, Extensible HyperText Markup Language (XHTML), Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), HyperText Markup Language (HTML), Java, JavaScript, PHP: Hypertext Preprocessor (PHP), Visual Basic (VB), Visual Basic Script Edition (VBScript) and Wireless Markup Language (WML) (Wikipedia, 2011c).

Adobe Dreamweaver CS5 software empowers designers and developers to build standards-based websites with confidence. Design visually or directly in code, develop pages with content management systems, and accurately test browser compatibility (Adobe, 2011).

2.4.2.2 Microsoft Visual Studio 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native codes together with managed code for all platforms. It supports various types of languages such as C/C++, VB.NET, C# and F#. Support for other languages too such as M, Python, Ruby, XML/XSLT, HTML/XHTML, JavaScript, CSS, ASP and ASP.NET (Wikipedia, 2011d).

Visual Studio 2010 is packed with new and enhanced features that simplify the entire development process from design to deployment. Customize your workspace with multiple monitor support. Create rich applications for SharePoint and the Web.

Target multiple versions of the .NET Framework with the same tool (Microsoft, 2011).

2.4.2.3 Netbeans IDE 6.9

The NetBean is an integrated development for Mac, Windows, Linux and Solaris which is award-winning. An open-source IDE and an application platform which consist in the NetBeans project enable developer to create web, desktop, enterprise and mobile applications with rapid speed by using the Java platform. (Netbeans, 2000).

2.4.2.4 Comparison of Advantages and Disadvantages of Integrated Development Environment (IDE)

The table below as shown in Table 2.4 consists of comparison of advantages and disadvantages of integrated development environment (IDE) between Adobe Dreamweaver CS5, Microsoft Visual Studio 2010 and Netbeans IDE 6.9.

Table 2.4: Comparison of Advantages and Disadvantages of Programming Tools

	Adobe Dreamweaver CS5	Microsoft Visual Studio 2010	Netbeans IDE 6.9
Software price	Not free	Not free	Free
Memory usage	High	High	Low
OS Compability	Any	Window only	Any
Server and client side scripting	Yes	Yes	Yes

Platform	Any	Microsoft platforms only	Java platforms only
Code coverage	Large	Limited	Limited
GUI	Good	Good	Good
System Requirement	High	High	Low

2.4.3 Database

A database consists of an organized collection of data for one or more uses, typically in digital form (Wikipedia, 2011e).

2.4.3.1 MySQL

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. MySQL is the world's most popular open source database software because of its consistent fast performance, high reliability and ease of use with over 100 million copies of its software downloaded or distributed throughout its history.

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX, giving you the kind of flexibility that puts you in control (MySQL.com, 2011)

2.4.3.2 Microsoft SQL Server 2008 R2

Microsoft SQL Server is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL. The code base for MS SQL Server originated in Sybase SQL Server, and was Microsoft's entry to the enterprise-level database market, competing against Oracle, IBM, and others.

SQL Server 2008 R2 adds certain features to SQL Server 2008 including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services (Wikipedia, 2011f).

2.4.3.3 Oracle Database 11i/11g

An Oracle database is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures (Oracle, 2011b).

2.4.3.4 Comparison of Advantages and Disadvantages of Database

The table below as shown in Table 2.5 consists of comparison of advantages and disadvantages of database between MySQL, Microsoft SQL Server R2 and Oracle Database 11i/11g.

Table 2.5: Comparison of Advantages and Disadvantages of Database

	MySQL	Microsoft SQL Server R2	Oracle Database 11i/11g
Software price	Free	Not free	Not Free
Simplicity	Yes	Yes	No
Access speed	Fast	Slow	Fast
Server side scripting language	Any	Any	Any
Platform	Any	Microsoft platforms only	Any
Security	Strong	Strong	Strong
GUI	No	No	Yes
Storage	Big	Big	Big

CHAPTER 3

METHODOLOGY

This chapter will be devoted to discuss the software process model for the system development and developmental issues.

3.1 Over View to the Model

3.1.1 The way to adopting V model in Iterative & Incremental development

Let consider incremental development at first, the project is divided to various independent parts by incremental development and the division is as much as possible. Assuming a example of my development of a society management website with parts like member registration, member login, view profile, view video, post comments and resource booking. Depending on the product owners requirements, I can start with member registration, member login and view profile by complete these and during this stage, adopting V- model inside incremental development, each phase in V-model is need to complete. Once these part are ready, then can

proceed to the next set. These all parts are also possible to working simultaneously and integrating them when ready in the central repository.

The combination of v-model and incremental development process would be like this:

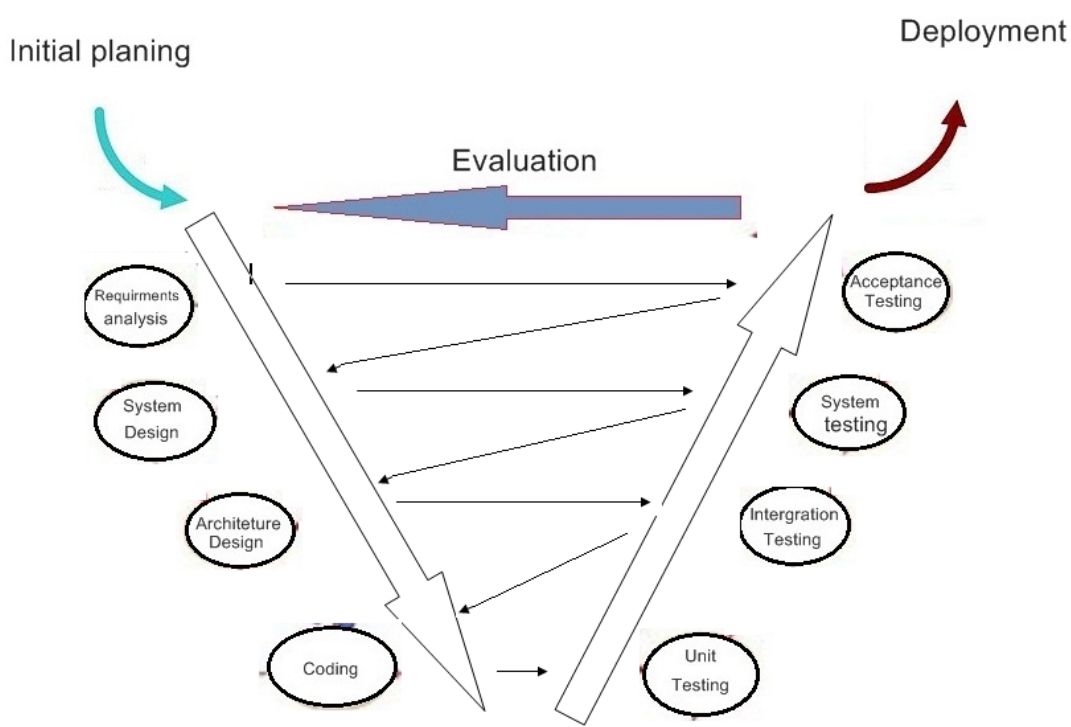


Figure 3.1: combination of V-model and iteration & incremental

From figure 3.1, the combination of V- model is not much different with previous original V –model, there are only a little changes such as initial planning , deployment when testing are done and evaluation for next iteration. In incremental process, developer needs to identify which of the services are most important and which are least important to them.

By using the same example, one way of looking at iterative development is to see the first iteration comprising of member registration, member login and view profile. The second iteration can comprise of photos, photos search, video, and video search. And a better way to look iteration would be like this:

- Provide member registration with form fields like name, email address, password and submit button.

- In the next “iteration” can divide name into first name and last name, email address and password to be confirmed twice and eliminate zip code entirely and add birth date to the registration.

3.2 Initial Planning

The new model that combined from V-model and iteration & incremental development are shown in figure 3.1. By using this model, the project is dividing into various independent parts by incremental development. The divided part will apply into v-model, but this V-model is different with original model which is changed to have iteration process. By using this method, developer can identify which of the service are most important and which are least important to them. Thus, developer can start a simple build and work around in initial set to gaining experience and knowledge, and iteration is applied to help learning advantage from build to build. From the result of the discussion, requirement of member of 24 seasons drums society of each model are listed below:

- xi. Manage donation
- xii. resource booking
- xiii. Resource availability
- xiv. Posting news and events
- xv. Add comments to news & events
- xvi. Manual performance/show timetable
- xvii. Picture gallery
- xviii. Search on photo
- xix. Attendance management
- xx. Management for society account
- xxi. Video gallery
- xxii. Search on video
- xxiii. Manage members profile
- xxiv. Request for show and performance
- xxv. Performance management
- xxvi. Society complaint management
- xxvii. Help and information

- xxviii. Self timetable
- xxix. Resource/asset management
- xxx. Portal management

Base on the requirement of member from 24 Season Drums society, the allocation of services to increments depends on the service priority. The highest priority services are delivered first to them. These 20 modules will be divided to various independent parts by incremental development which is 6 divisions, the priority service of these incremental set also arranged in the order which the first set is the highest priority services, the sets of incremental are shown below:

First incremental set-

- i. Manage members profile
- ii. Portal management
- iii. Attendance management
- iv. Add comments to news & events
- v. Posting news and events

Second incremental set:-

- i. resource booking
- ii. Resource availability
- iii. Resource/asset management

Third incremental set:-

- i. Manual performance/show timetable
- ii. Request for show and performance
- iii. Performance management
- iv. Self timetable

Fourth incremental set:-

- i. Picture gallery
- ii. Search on photo
- iii. Video gallery
- iv. Search on video

Fifth incremental set:-

- i. Manage donation
- ii. Society complaint management
- iii. Help and information
- iv. Management of Society Account

3.3 Verification Phases

Each of the incremental set, all of these module will following the stage of V-model and the iteration process will keep on iterate until the full system is implemented. For the new V-model, there is having 2 different phases which is Verification phases and Validation Phases. For Verification Phase there is having requirement analysis system design, architecture design and module design.

3.3.1 Requirements analysis

This is the first step in the verification process. It is in here that decide the incremental set and its function. During this stage the developers are not going to discuss how it is going to be built but discussion and a user requirement is needed. The function of the incremental set, performance, security, data, interface and needed in this document. The requirement will be identify by following seperated incremental set.

As stated, the 24 Season drums Society management System requirements for each incremental set are listed as below:

First incremental set-

- Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the member profile.
- Members will provide an ID and password to log in to the website.
- Members can update their profile manually.
- To record the attendance for each training or activity.
- To provide a post comment space for 24 season drums society members to post their comments to share their opinion.
- To provide a site to let admin or members to post the latest information and events

Second incremental set:-

- To provide a form fill-in site to let 24 season drums society members apply event/tools.
- To provide a form fill-in site to let members request for venue booking for training or practice.
- Check for latest condition of tools and record.
- Admin has to update for quantity and quality of tools.

Third incremental set:-

- Check for latest condition of tools and record.
- Admin has to update for quantity and quality of tools.
- Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the performance and show information.
- Provide a calendar that full of event schedule to let members and open users to view all the upcoming events.
- Open user can request for show and performance by key in venue and time to waiting for approval.
- Admin has to check for availability for performance date and discuss with group members by posting new.

- Admin has to reject or approve the request once have an answer for discussion.
- Admin has to list down all preparation for each upcoming performance such as uniform and tools need to use.
- Manage for transportation when needed.

Fourth incremental set:-

- Provide search function to let users search wanted video by key in the related information such as members name, venue and name of performance.
- Admin able to update the video gallery by category to several albums.
- Members and open users can view the video that recorded from performance and others activity.
- Admin able to add in the information of the video such at time, performance venue and name of members in photo.
- Provide search function to let users search wanted photo by key in the related information such as members name, venue and name of performance.
- Admin able to update the Picture gallery by category to several albums.
- Members and open users can view the photos that snap from performance and others activity.
- Admin able to add in the information of the photo such at time, performance venue and name of members in photo

Fifth incremental set:-

- to provide a site to let open users or members to get more information about donation
- Admin has to manage the donator information such as their personal profile information.

- Admin has to update account balance and make a confirmation once transaction was successful.
- Provide a site to let public user to give a donation or sponsor.
- To provide a site to let users to make a complaint for anything that related to society to improve the efficiency.
- Provide help and information to solve user problem when using this website.
- Introduce about 24 season drum society to let all user more understand the purpose of 24 drums performance.
- To manage the member fees and other spent cost.
- To record all the spent cost and income fees.

A brief documentation about the system requirements has sent to the 24 Season Drums Society to refer after double confirmation of all the requirement details. After that, further analysis and research of the system requirements has compared with the existing system on the market today. Lastly, a schedule planning from the beginning until the end of the system development has drafted in the Gantt chart which shown in Appendix A. The details of requirement documented in Software Requirement Specification (SRS) as an external documentation.

3.3.2 System Design

System design is the phase where system engineers analyze and understand the properties of the system by studying the user requirements document.

The software specification document which serves as a blueprint for development phase is generated. This document contains the general system organization, menu structures, data structures. Other technical documentation like entity diagrams, data dictionary will also be produced in this phase. The documents for system testing are prepared in this phase.

3.3.2.1 Entity-Relationship Diagram

Entity-Relationship diagram (ERD) is used to design the database of the system. It was chosen as a model to show the relationships between different entities in the database. 24 Season Drums Society Management System consists of a database which contains data for one or more users. For database that store member profile are link to many others database to identify the member identity during proceed to next progress Member profile are link to attendance database which to manage the attendance database by each of the Attendance list can be manage by zero to many member profile. In contrast, each of the member profile also can be state by zero to many attendance lists. For portal management, there are has 1 to many member profile for related to post comments and reply comments, and each portal can be post by zero to many members. Each resource can be reserve by one member and each member can have more than one or zero reserved resource. Each resource availability is control by zero to many resources list and each resource availability has zero to many resources list. Resources availability also limited by resource condition by one to one resource.

Each of the member timetable also manage by one member and each member only have one member timetable and each member timetable arranged by one to many performance timetable. Performance request, account, donation and complaint database are independent database. ERD of the system is shown in Figure 3.2.

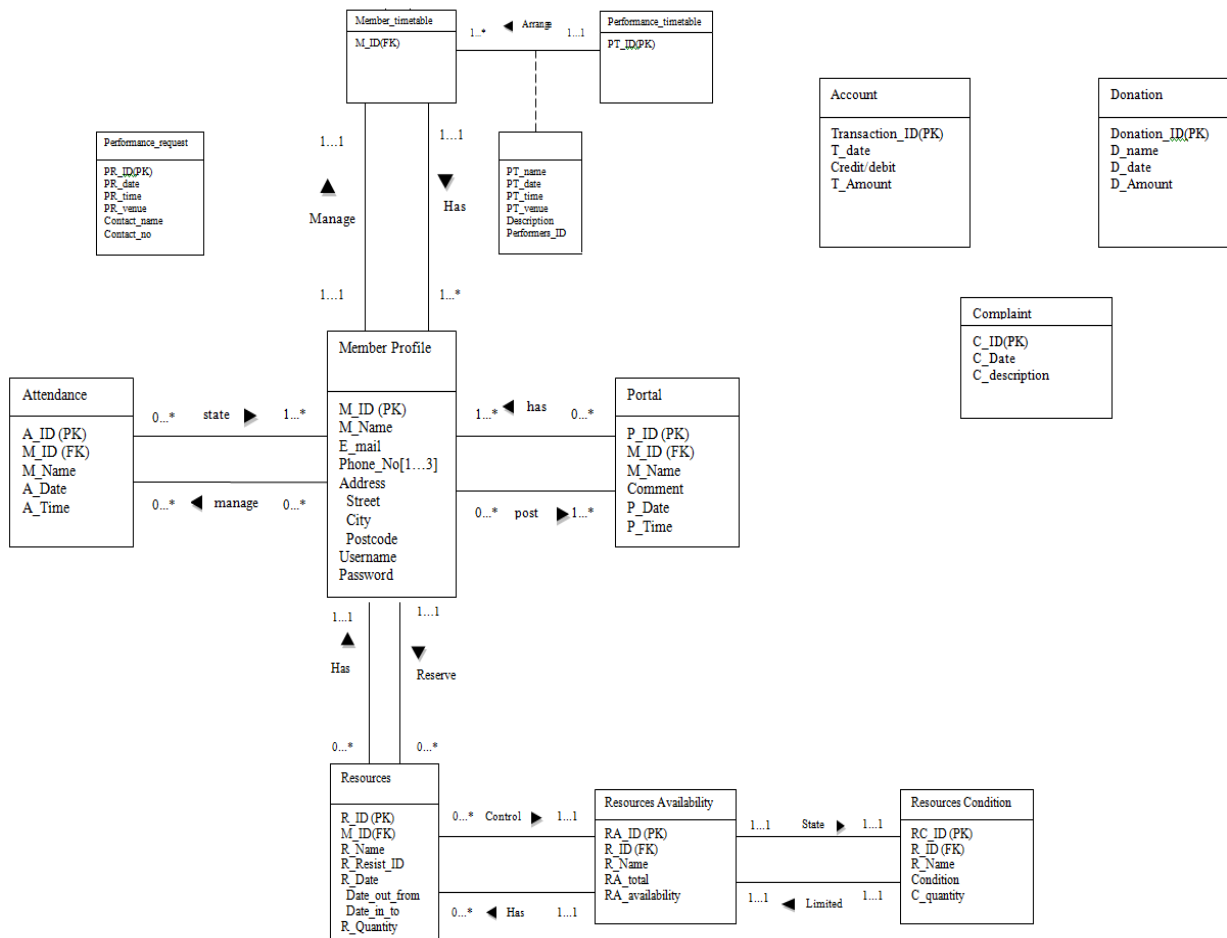


Figure 3.2: Entity-Relationship diagram (ERD) for whole system

The ERD will break to five small independent part of ERD according to the incremental set that has been plan. That 5 ERD are shown in figure 3.3 until figure 3.7.

First incremental set

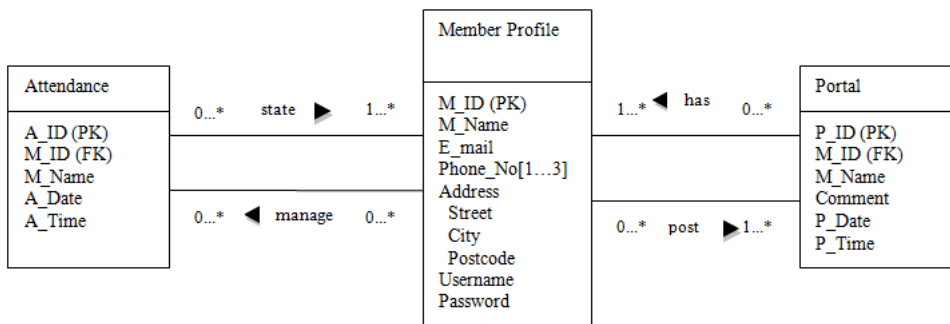


Figure3.3 Entity Diagrams for first incremental set

Second incremental set

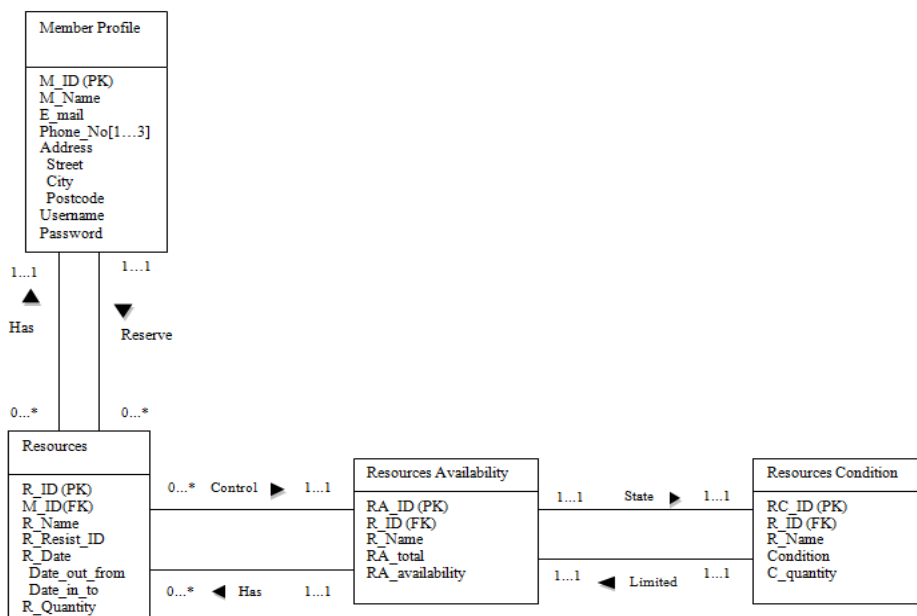


Figure3.4 Entity Diagrams for second incremental set

Third incremental set

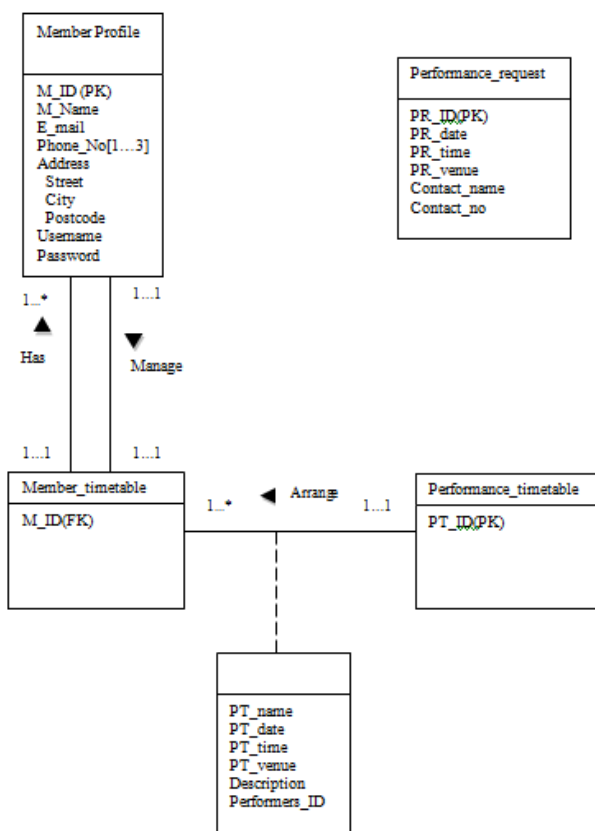


Figure3.5 Entity Diagrams for third incremental set

Fourth incremental set

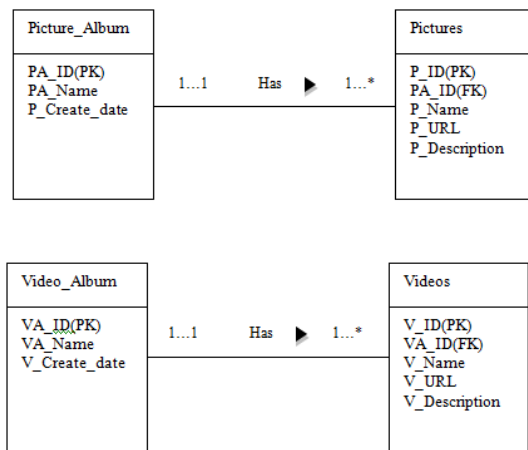


Figure3.6 Entity Diagrams for fourth incremental set

Fifth incremental set

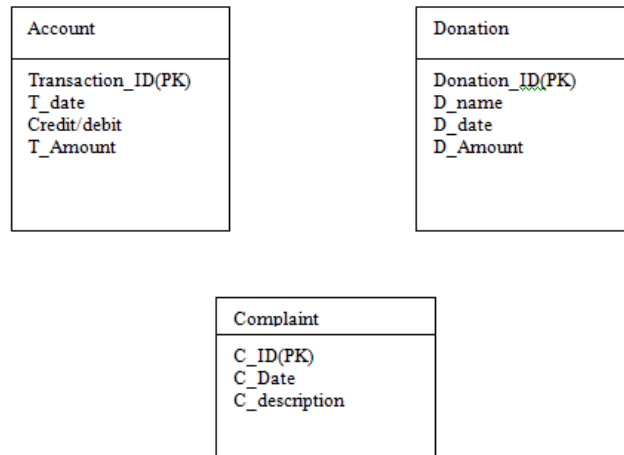


Figure3.7 Entity Diagrams for fifth incremental set

3.3.3 Architecture Design

Architecture Design consists of the list of modules, brief functionality of each module, interface relationships, dependencies, database tables, architecture diagrams.

3.3.3.1 List of module and functionality

1. Manage donation

- to provide a site to let open users or members to get more information about donation
- Admin has to manage the donator information such as their personal profile information.
- Admin has to update account balance and make a confirmation once transaction was successful.

- Provide a site to let public user to give a donation or sponsor.

2. resource booking

- To provide a form fill-in site to let 24 season drums society members apply event/tools.
- To provide a form fill-in site to let members request for venue booking for training or practice.

3. Resource availability

- Check for latest condition of tools and record.
- Admin has to update for quantity and quality of tools.

4. Posting news and events

- To provide a site to let admin or members to post the latest information and events

5. Add comments to news & events

- To provide a post comment space for 24 season drums society members to post their comments to share their opinion.

6. Manual performance/show timetable

- Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the performance and show information.
- Provide a calendar that full of event schedule to let members and open users to view all the upcoming events.

7. Picture gallery

- Admin able to update the Picture gallery by category to several albums.
- Members and open users can view the photos that snap from performance and others activity.

- Admin able to add in the information of the photo such as time, performance venue and name of members in photo

8. Search on photo

- Provide search function to let users search wanted photo by key in the related information such as members name, venue and name of performance.

9. Attendance management

- To record the attendance for each training or activity.

10. Management for society account

- To manage the member fees and other spent cost.
- To record all the spent cost and income fees.

11. Video gallery

- Admin able to update the video gallery by category to several album.
- Members and open users can view the video that recorded from performance and others activity.
- Admin able to add in the information of the video such as time, performance venue and name of members in photo.

12. Search on video

- Provide search function to let users search wanted video by key in the related information such as members name, venue and name of performance.

13. Manage members profile

- Admin has to use management processes such as move/add/change, procurement, storage, and disposal to manage the member profile.
- Members will provide an ID and password to log in to the website.

- Members can update their profile manually.

14. Request for show and performance

- Open user can request for show and performance by key in venue and time to waiting for approval.
- Admin has to check for availability for performance date and discuss with group members by posting new.
- Admin has to reject or approve the request once have an answer for discussion.

15. Performance management

- Admin has to list down all preparation for each upcoming performance such as uniform and tools need to use.
- Manage for transportation when needed.

16. Society complaint management

- To provide a site to let users to make a complaint for anything that related to society to improve the efficiency.

17. Help and information

- Provide help and information to solve user problem when using this website.
- Introduce about 24 season drum society to let all user more understand the purpose of 24 drums performance.

18. Self timetable

- Let members can manage their own schedule by themselves such as view their own involved performance date and time.

19. Resource/asset management

- Management of resources for administrator

20. Portal management

- Management of portal for administrator

3.3.3.2 Workflow Design

Workflow of the system and a sequence of operations were represented by using a flow chart that will show the steps as boxes of various kinds, and their order by connecting these with arrows. This diagrammatic representation can give a step-by-step solution to a given problem. The flow chart of the system for future and existing member for society is shown in Figure 3.8 and the flow chart of the each incremental set is shown from figure 3.9 until figure 3.13

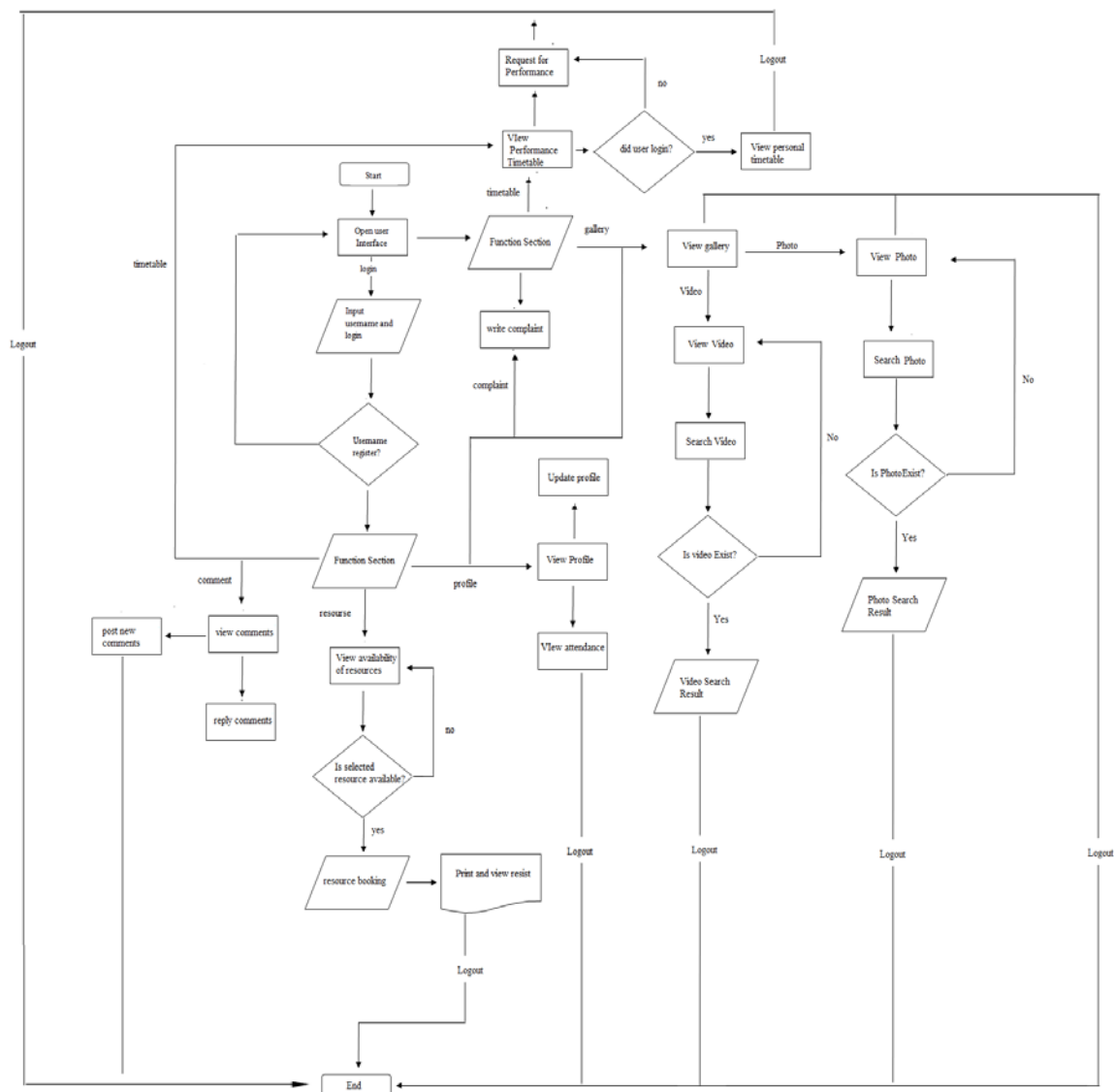


Figure 3.8 flow chart of 24 season drums society for existing member, future member and open user

First incremental set

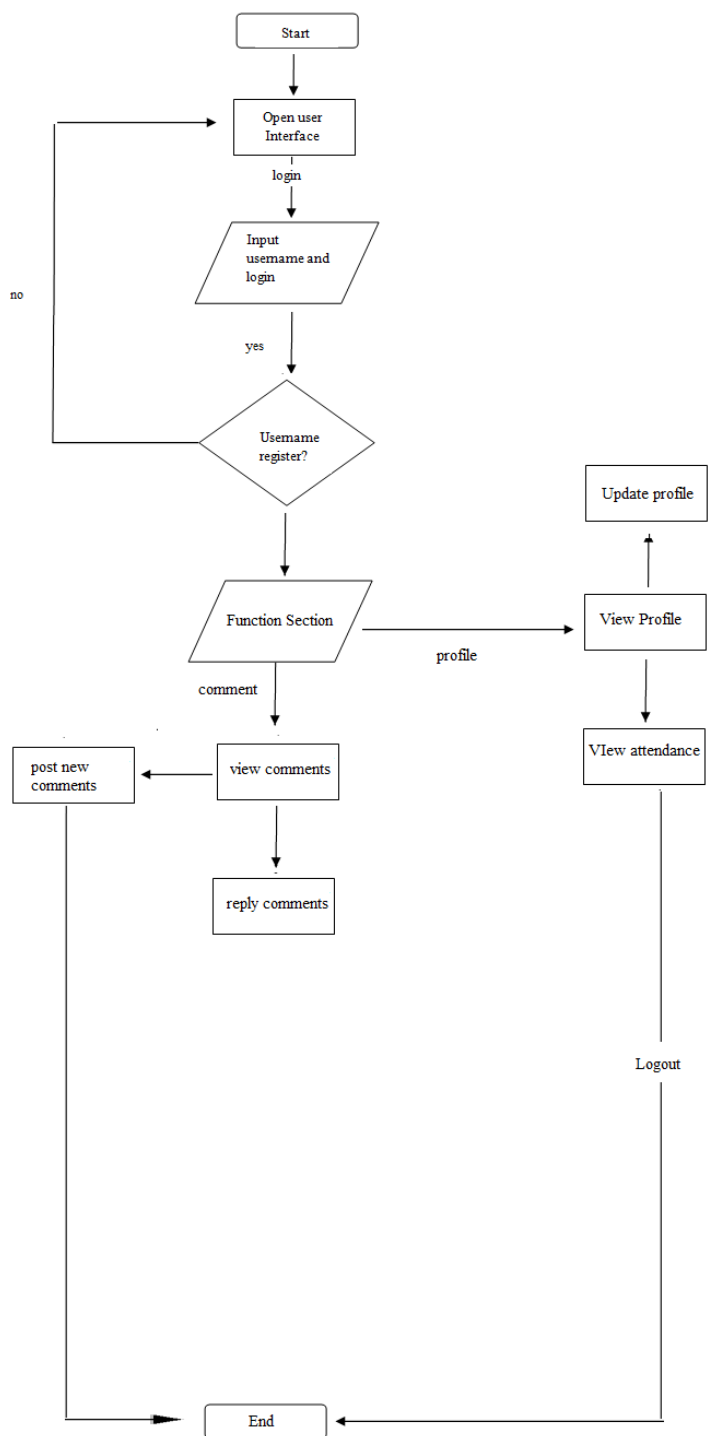


Figure 3.9 flow chart for first incremental set

Second incremental set

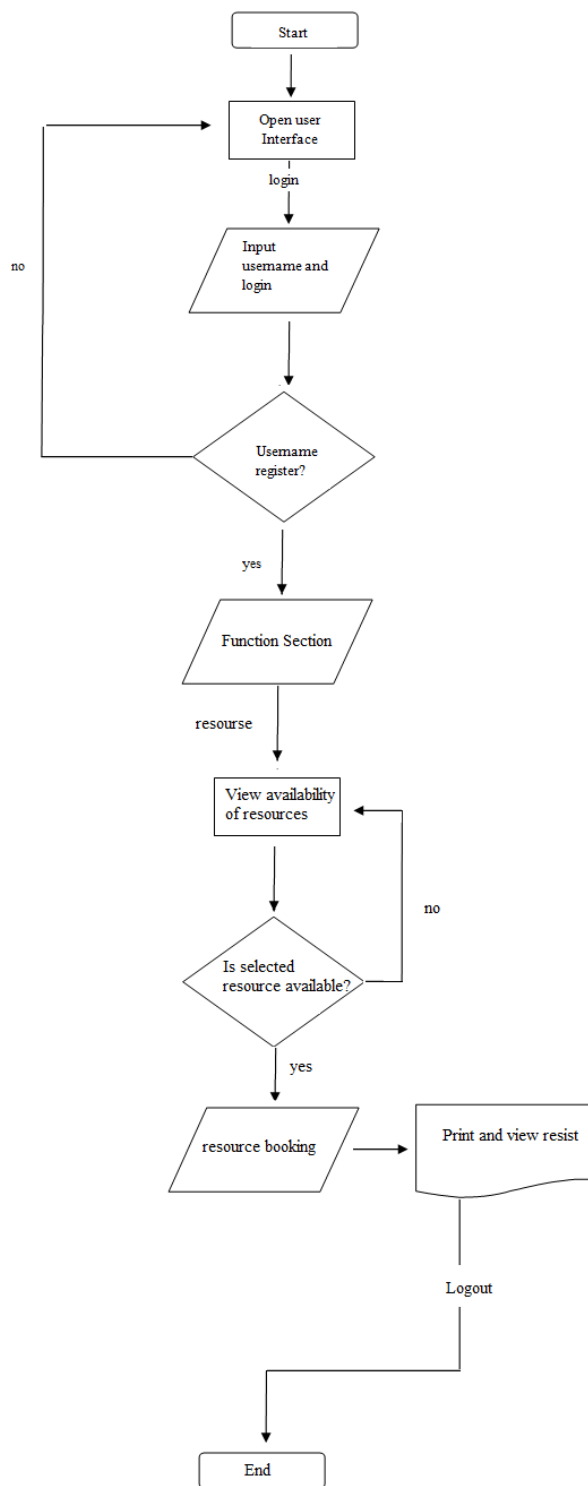


Figure 3.10: flow chart for second incremental set

Thrid incremental set

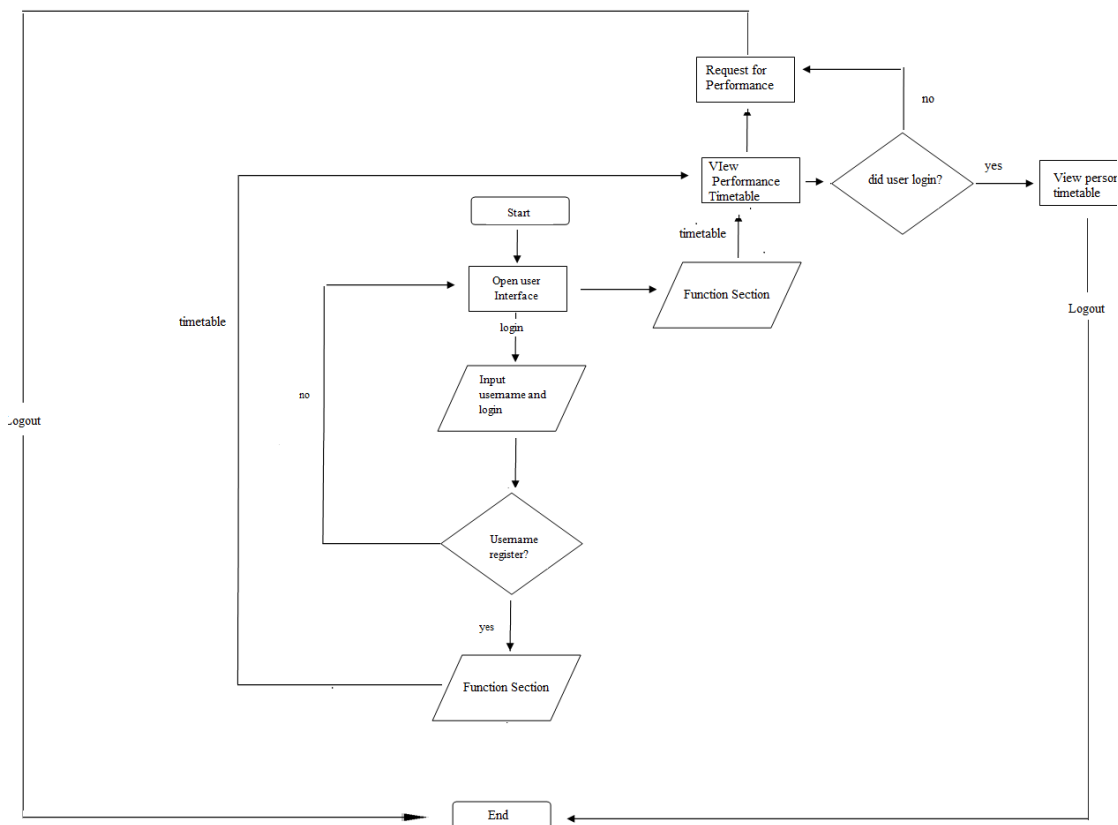


Figure 3.11 flow chart for third incremental set

Fourth incremental set

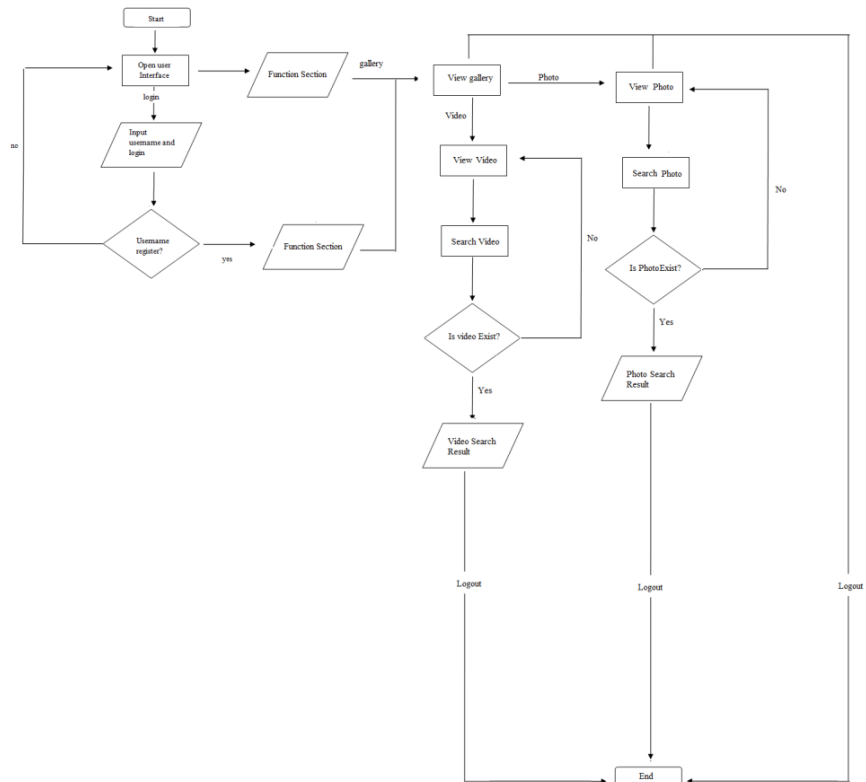


Figure 3.12: flow chart for fourth incremental set

Fifth incremental set

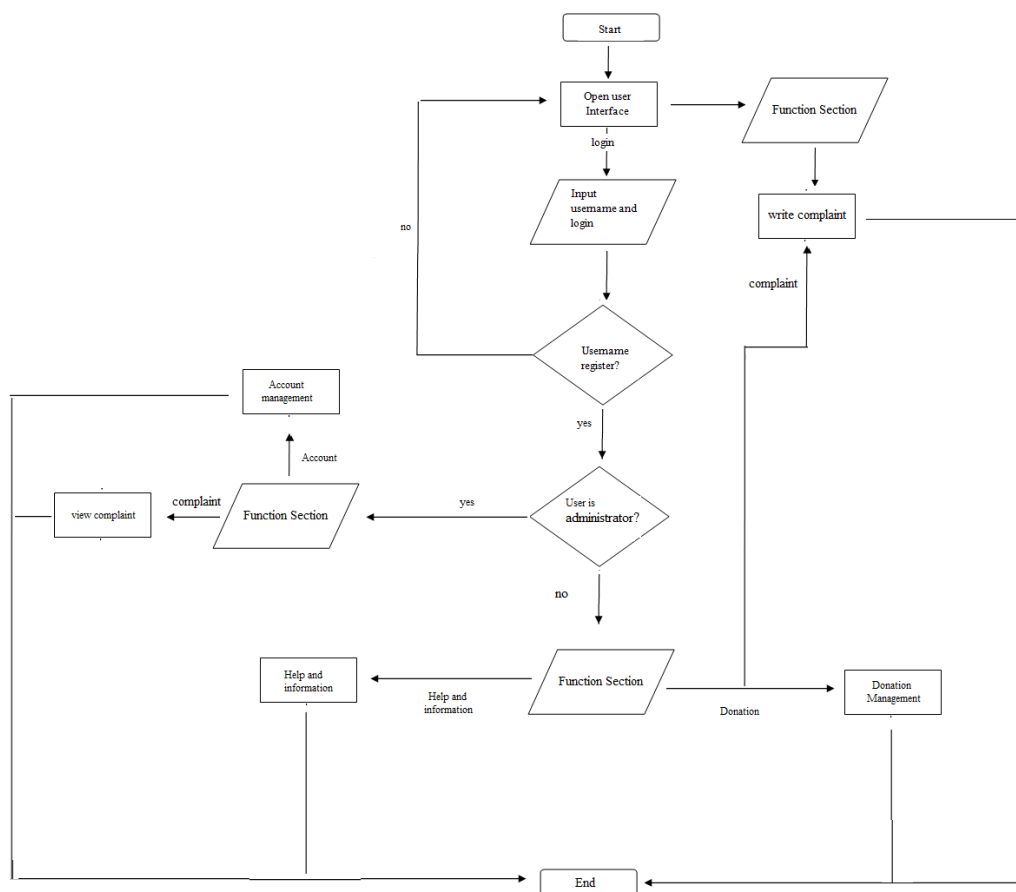


Figure 3.13: flow chart for fifth incremental set

3.3.3.3 Interface Design

Interface design focus on the user's experience and interaction with the system. User interface-flow diagram was chosen to model the interactions that users have with the system. The user interface flow diagram for future and existing members for society is shown in Figure 3.14.

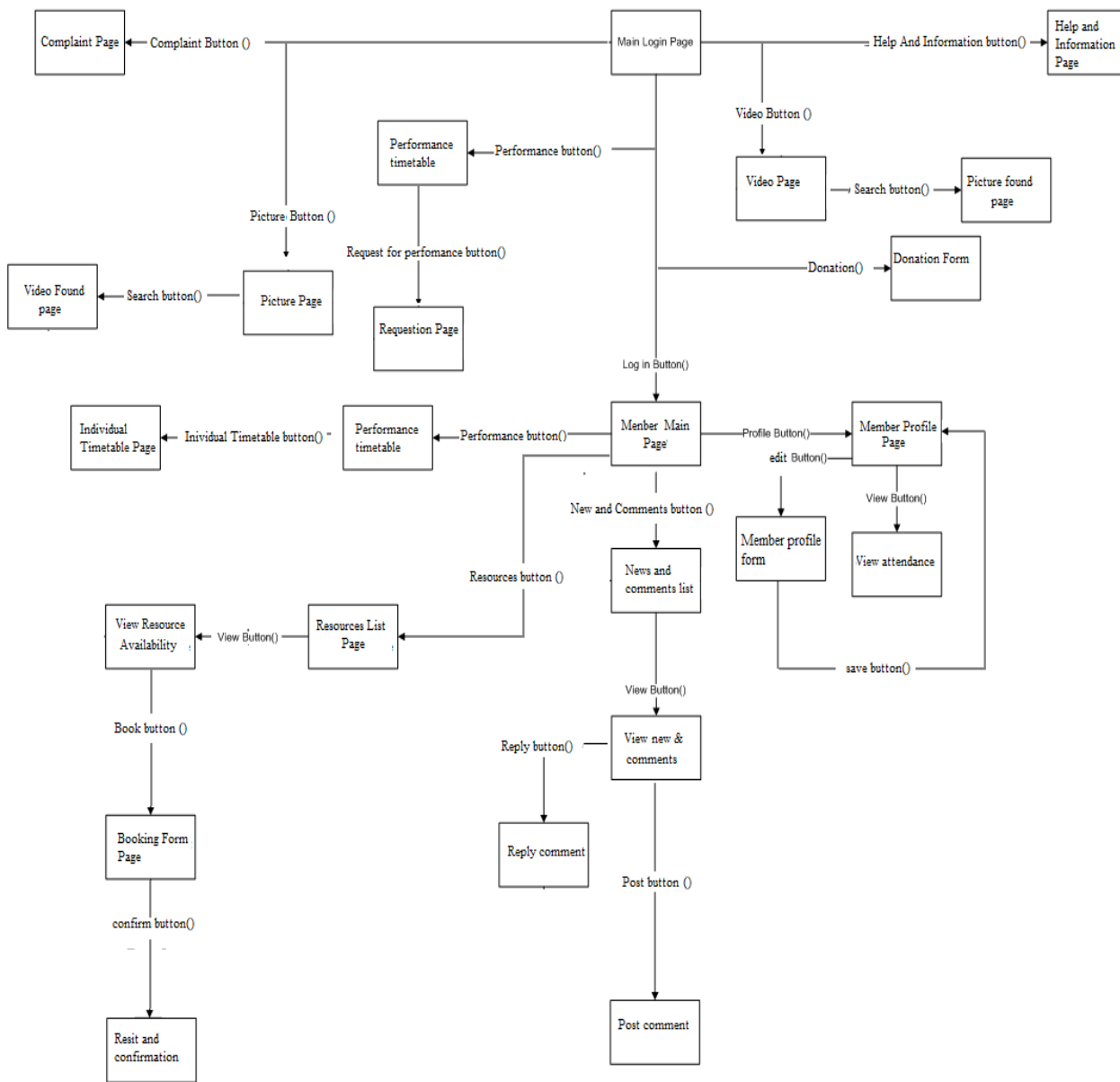


Figure 3.14: User Interface Flow Diagram for 24 Season drums society

3.3.3.4 Data Flow Design

Context diagram was chosen as a process model to represent all the external actors that are interacting with the system. It is the overall representation of the whole system which will also shows all the inputs and outputs taking place with all

the external actors within the system. The context diagram of the system is shown in Figure 3.15. Data flow diagram (DFD) is commonly used in structured system design. It is chosen as a process model to show all the data flow from external entities into the system, showed how data moved from one process to another and also showing all the storage and data store available in the system. Level 0 data flow diagram is chosen to show a more detail of data flow and processes happening in the system. The DFD level 0 of the system for future and existing members of Society is shown in Figure 3.16, and the open user of system is shown in Figure 3.17, while the DFD level 0 of the system for administrator of system is shown in Figure 3.18.

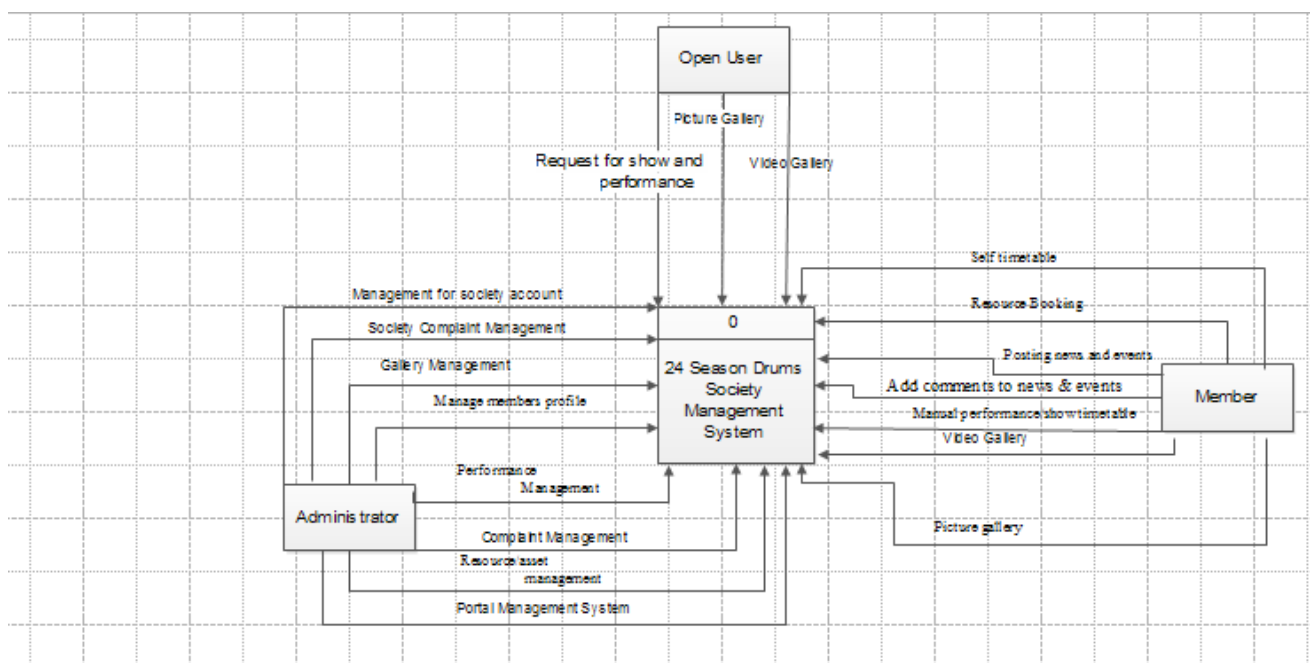


Figure 3.15: Context Diagram of 24 Season Drums Society Management System

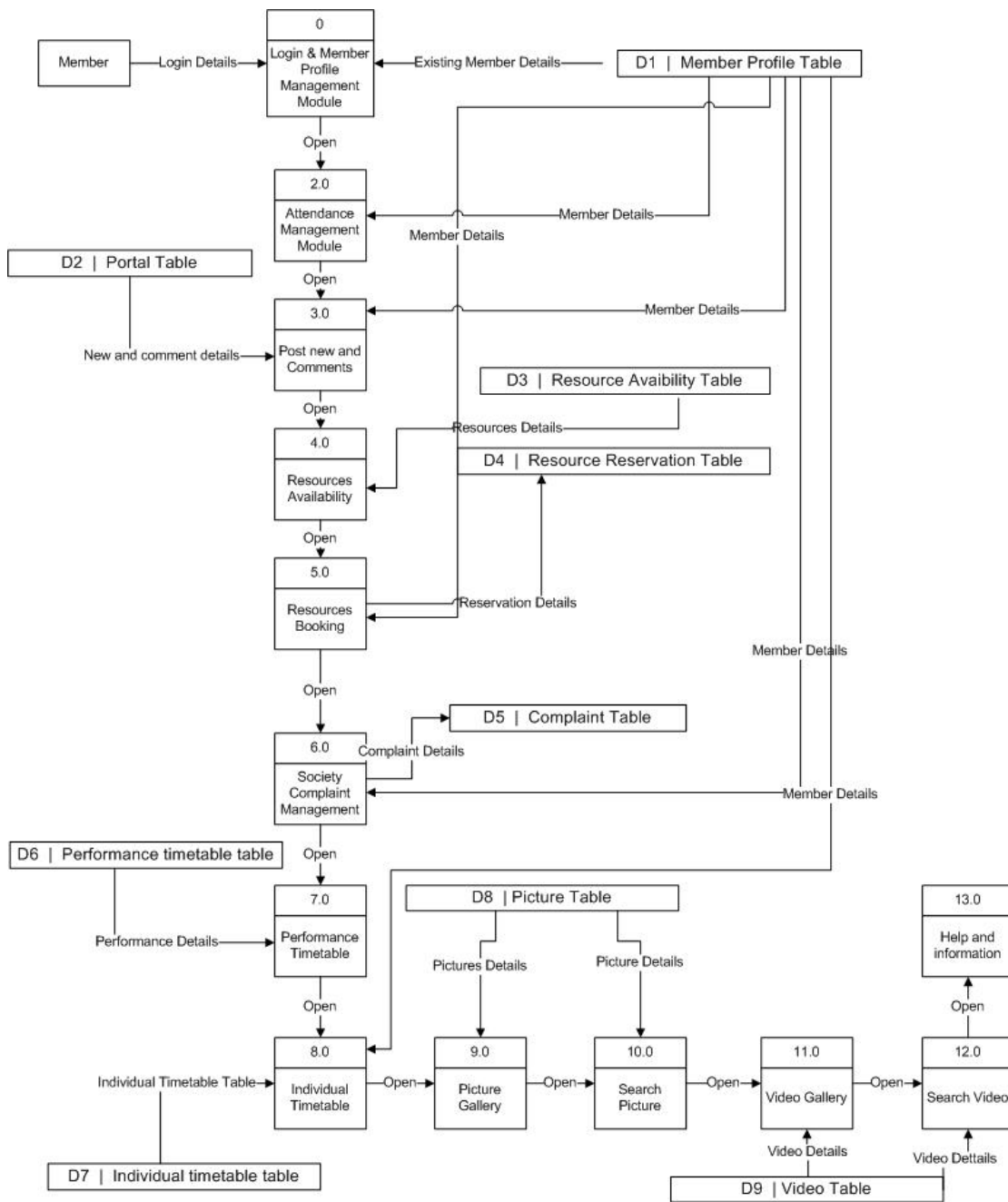


Figure 3.16: DFD Level 0 of 24 Season Drums Society Management System for existing member

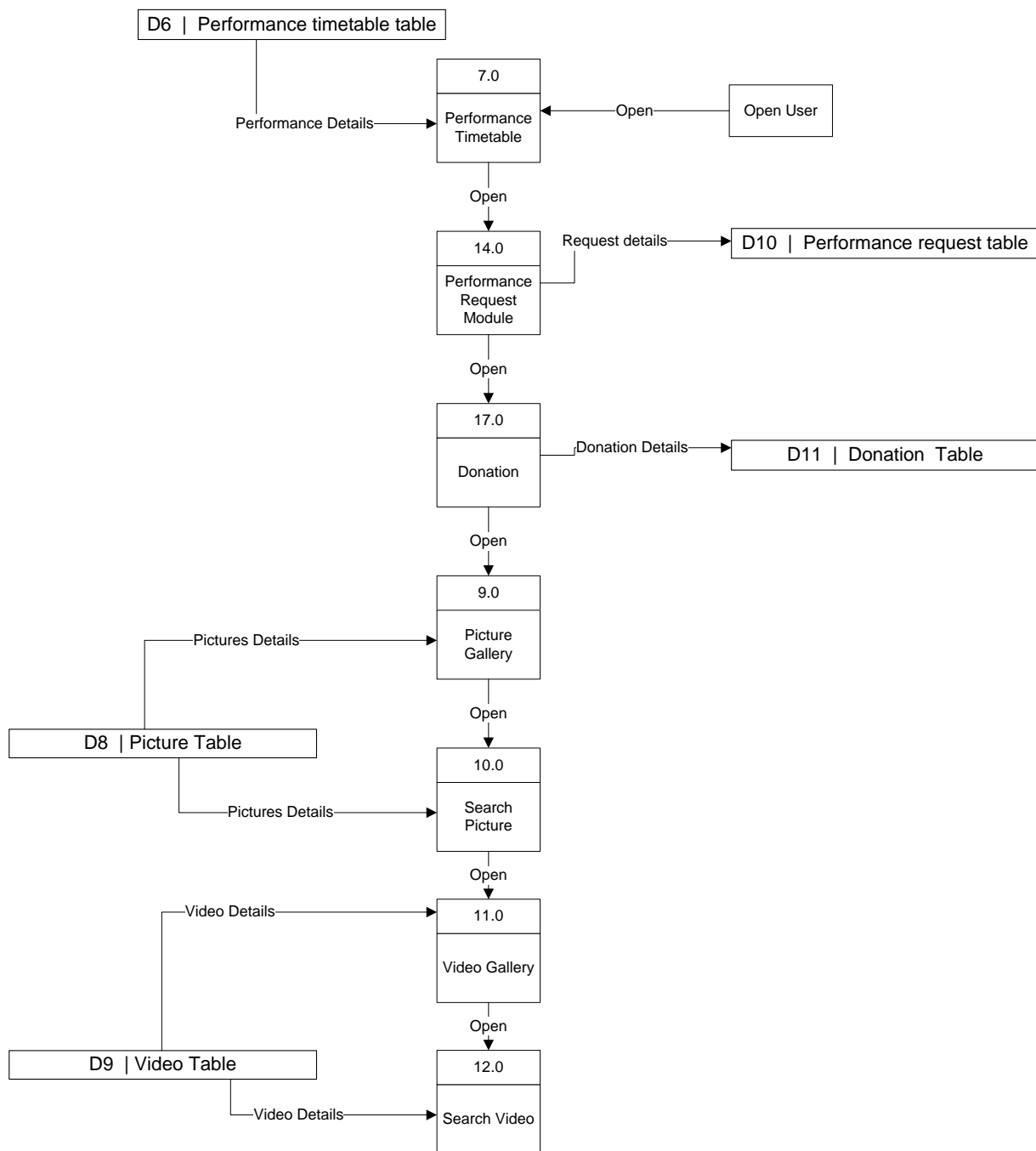


Figure 3.17: DFD Level 0 of 24 Season Drums Society Management System for open user

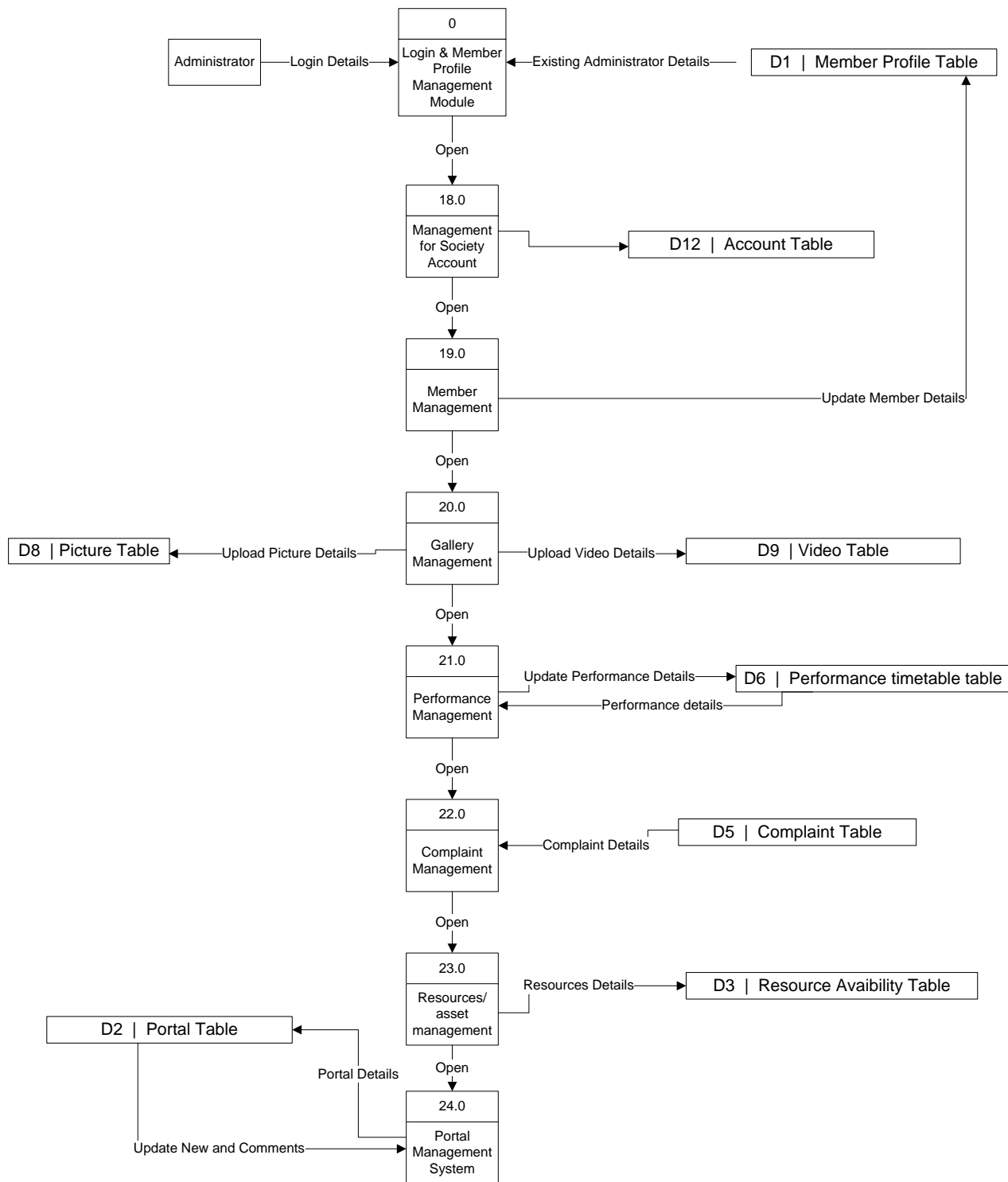


Figure 3.18: DFD Level 0 of 24 Season Drums Society Management System for administrator

3.4 Validation Phases

iv. Unit Testing

- Unit tests are created by programmers or occasionally by white box testers.
- The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage.

v. Intergration Testing

- In intergration testing the separate modules will be tested together to expose faults in the interfaces and in the intraction between intergrated components.
- Testing is usually black box as the code is not directly checked for errors.

vi. System testing

- System testing will compare the system specifications against the actual system.

3.5 Software and Hardware Specification

Several software and hardware have been used to in the development of Online Industrial Sales and Support System. Hardware is the physical aspect of the computer, while software is a general term or part for the various kinds of program used to operate computer and other devices.

3.5.1 Software Specification

After an in depth research, ASP.NET is chosen as the programming language in developing the system. So the development tools are selected based on the chosen language environment for the development of 24 Season drums society after comparing all the possible tools. Table 3.1 below shows the list of software that is used during the development of the system.

Table 3.1 Software Specification for 24 Season drums society

Software	Purpose
Microsoft Visual Studio 2010	System interface design and coding implementation
MySQL	Database platform
Apache HTTP Server	Web server platform
Microsoft Office -Microsoft Office Word 2007 - Microsoft Office Power Point 2007 - Microsoft Office Project 2007 - Microsoft Office Visio 2007	Documentation, presentation slide, schedule planning and graphical diagram design
WinRAR 3.93	File compress and decompress
Foxit Reader 5.0.2.0718	Reading pdf files
Google Chrome 14.0.835.202 m	Browse and test system
Window 7 Professional 32bit Operating System	Operating system used in development
Avast! Pro Antivirus 5.0545	Protect from virus and spyware

3.5.2 Hardware Specification

The hardware specification for the system is based on the system requirements of the chosen development tools. Table 3.2 below shows the list of hardware that is used during the development of the system.

Table 3.2 Hardware Specification for Online Industrial Sales and Support System

Hardware	Specification	Purpose
Laptop	-Intel(R) Core(TM)2 Duo CPU T6600 @ 2.26GHz -2GB RAM -250 GB Hard Disk	To run the operating system for development
Printer	brother DCP-J125 Printer	Print out the documentation
External Hard Disk	Buffalo HD-PXT320U2	Backup data
USB Flash Disk	Kingston 4 GB Data Traveler	Transferring data
CD	700 MB 48x	System backup and delivery

CHAPTER 4

IMPLEMENTATION

This chapter will be devoted to the implementation of database, interface and coding into the system.

4.1 Database Implementation

The 24 Seasons Drum Society Management System (24SDSMS) needs database to store the data for the society to manage the member profile and other functions. 24 Seasons Drum Society Management System is using phpMyAdmin as the database management system. It has 18 tables in the database named cb09041 as shown in Figure 4.1.

Table	Action	Rows	Type	Collation	Size	Overhead
advertisement	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 K1B	-
album	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 K1B	-
attendance_list	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 K1B	-
comment	Browse Structure Search Insert Empty Drop	20	InnoDB	latin1_swedish_ci	16.0 K1B	-
complaint	Browse Structure Search Insert Empty Drop	23	InnoDB	latin1_swedish_ci	16.0 K1B	-
donation	Browse Structure Search Insert Empty Drop	13	InnoDB	latin1_swedish_ci	16.0 K1B	-
event_training	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 K1B	-
help	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 K1B	-
main_page	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 K1B	-
member_profile	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 K1B	-
menu	Browse Structure Search Insert Empty Drop	14	InnoDB	latin1_swedish_ci	16.0 K1B	-
new	Browse Structure Search Insert Empty Drop	42	InnoDB	latin1_swedish_ci	16.0 K1B	-
performance	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 K1B	-
performer	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 K1B	-
pictures	Browse Structure Search Insert Empty Drop	27	InnoDB	latin1_swedish_ci	16.0 K1B	-
resources	Browse Structure Search Insert Empty Drop	25	InnoDB	latin1_swedish_ci	16.0 K1B	-
resourceslist	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 K1B	-
videos	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16.0 K1B	-
18 tables	Sum	221	InnoDB	latin1_swedish_ci	288.0 K1B	0 B

Figure 4.1: 24SDSMS Database

4.1.1 Database and Server Connection

The 24SDSMS need a server to host and link to the database. 24SDSMS is using Apache Server as local host and using PHP and MySQL code to connect the system with the server and database as shown in Appendix B-B1.

4.2 System implementation by using combination of V-model and iteration & incremental

Base on the requirement of member from 24 Season Drums society, the allocation of services to increments depends on the service priority. The highest priority services are delivered first to them. These 20 modules will be divided to various independent parts by incremental development which is 6 divisions, the

priority service of these incremental set also arranged in the order which the first set is the highest priority services, the sets of incremental are shown below:

First incremental set-

- i. Manage members profile
- ii. Portal management
- iii. Attendance management
- iv. Add comments to news & events
- v. Posting news and events

Second incremental set:-

- i. resource booking
- ii. Resource availability
- iii. Resource/asset management

Third incremental set:-

- i. Manual performance/show timetable
- ii. Request for show and performance
- iii. Performance management
- iv. Self timetable

Fourth incremental set:-

- i. Picture gallery and management
- ii. Search on photo
- iii. Video gallery and management
- iv. Search on video

Fifth incremental set:-

- i. Manage donation
- ii. Society complaint management
- iii. Help and information
- iv. Management of Advertisement

4.3 System Interface and Implementation

System interface is the key point of interaction between user and the system. The 24SDSMS interface is designed using CSS and HTML code with Adobe Dreamweaver CS5 while the 24SDSMS module function is using PHP and MySQL codes with Adobe Dreamweaver CS5.

4.3.1 Main Page / Login

The main page design frame with logo and banner, header, footer and navigation bar will be the same in other page. The main page has brief introduction on what is this system all about. The main page has navigation bar to link to picture, video performance, complaint, donation, help and information which all of them are the basic information of the company. Login form also shows on main page for user and administrator to login. The main page has navigation bar to link to the function provided by the system too as shown in Figure 4.2.

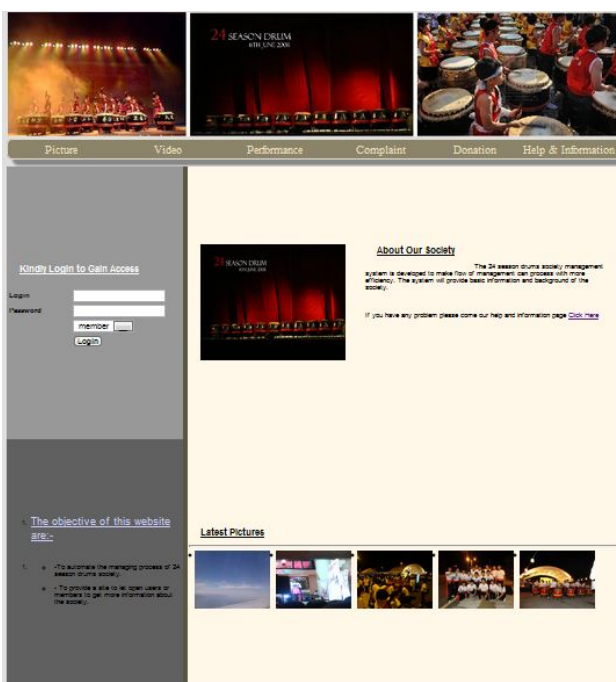


Figure 4.2: 24SDSMS Main Page

4.3.2 Member Profile – Member

This module is member profile which shows the information of member. This module provides edit function for member to edit and also upload profile picture. Member recently post also show on bottom of member information.

The screenshot displays the member profile interface. At the top, there are three banner images: a stage performance, a drum set with the text '24 SEASON DRUM 6TH JUNE 2006', and a group of drummers. Below the banners is a navigation bar with 'Logout', 'Welcome yong tat', and a search box containing 'members'. A left sidebar menu lists various options: Member Profile, Attendance List, Resources List, News and Comments, Post News, Performance Timetable, Individual Timetable, Picture Gallery, Video Gallery, Complaint Form, Donation Form, Help & Information, My Reservation History, and Advertisement. The main content area is titled 'Member Profile' and includes a 'Click here to go back' link. Under 'Profile Picture', there is a photo of a person with glasses drinking from a cup. The 'Profile Information' section lists the following details:

First Name	yong tat
Last Name	sum
email	kono_yakuza@hotmail.com
Phone Number	+6016-3917794
Address / Street	no2, jalan oempaka
City	ampang
Postcode	68000
Matrics ID	ob09041

An 'Edit' link is located below the profile information. The 'Member Recently Post' section shows three posts:

- Time/Date : 03-05-12 / 21:32:07
hello
yong tat
Comments
- Time/Date : 03-05-12 / 18:24:22
testing
yong tat
Comments
- Time/Date : 03-05-12 / 18:15:48
testing
yong tat
Comments

Figure 4.3: 24SDSMS Member profile – member

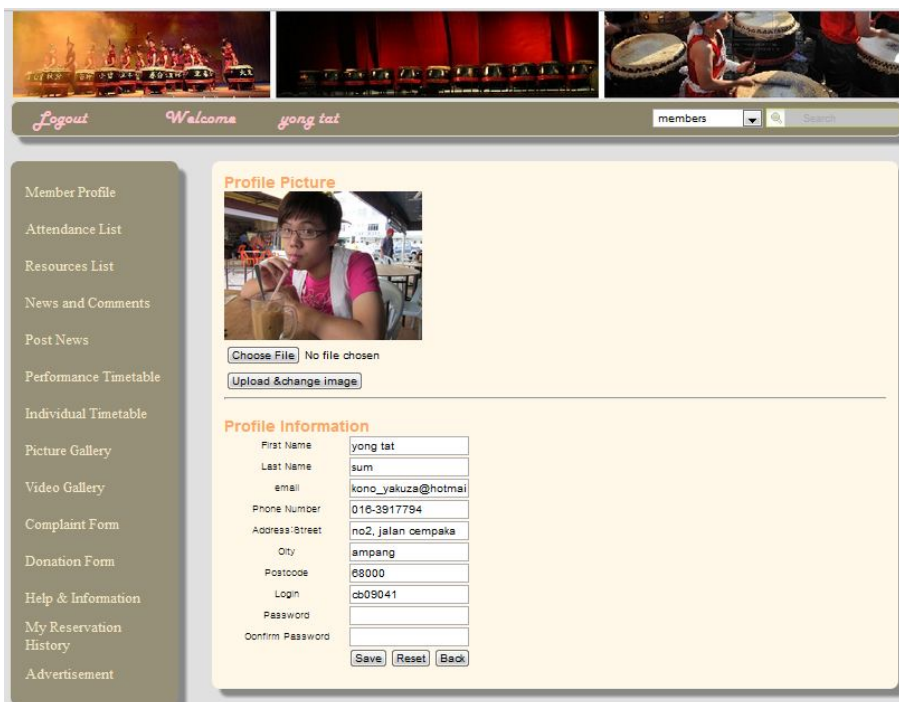


Figure 4.4 24SDSMS Edit Member profile – member

4.3.3 Attendance – Member

This module shows attendance of user according to event and training . This function also available for member to view event attended members.

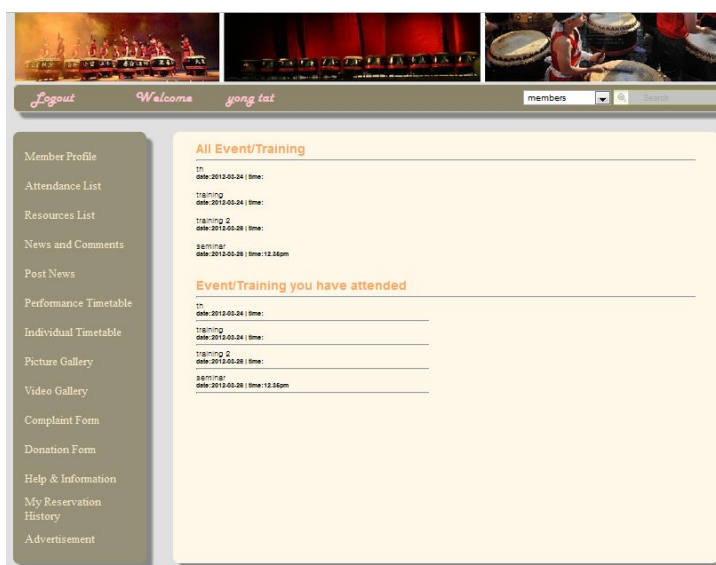


Figure 4.5: 24SDSMS Attendance List - Member

4.3.4 Resources Reservation

This module enable member to choose resources to reserve. The table will show the availability and quantity of resources. Once clicked in, member require to fill in reserve period and also quantity.

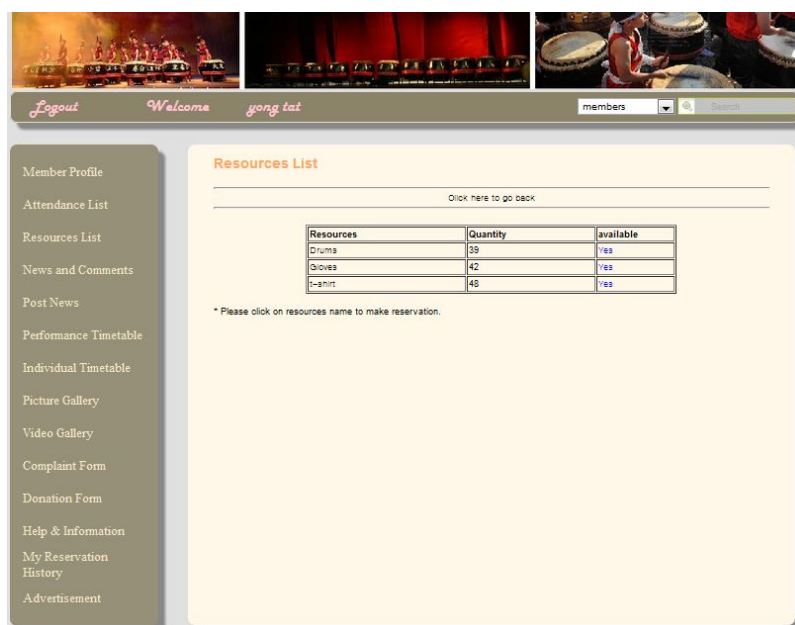


Figure 4.6: 24SDSMS Resources Reservation – Member

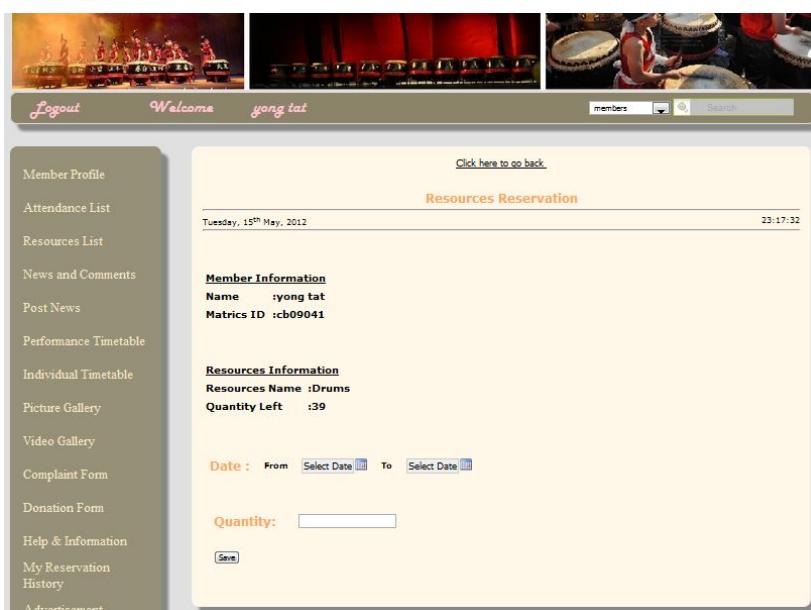


Figure 4.7: 24SDSMS Resources Reservation – Member



Figure 4.8: 24SDSMS Resources Reserve confirmation – Member

4.3.5 News and Comments

This module provides the capability for member and admin to interact with other user in the post new and comments as shown in Figure 4.9. Member and admin can view the list of topic in the selected thread as shown in Figure 4.9. Member and admin can also add a new topic for that particular thread as shown in Figure 4.11. Member and admin can view the list of replies in the selected topic as shown in Figure 4.10. Member and admin can also reply for that particular topic as shown in Figure 4.10. The coding implementation for add new topic function is shown in Appendix B7 and add new reply function is shown in Appendix B8.

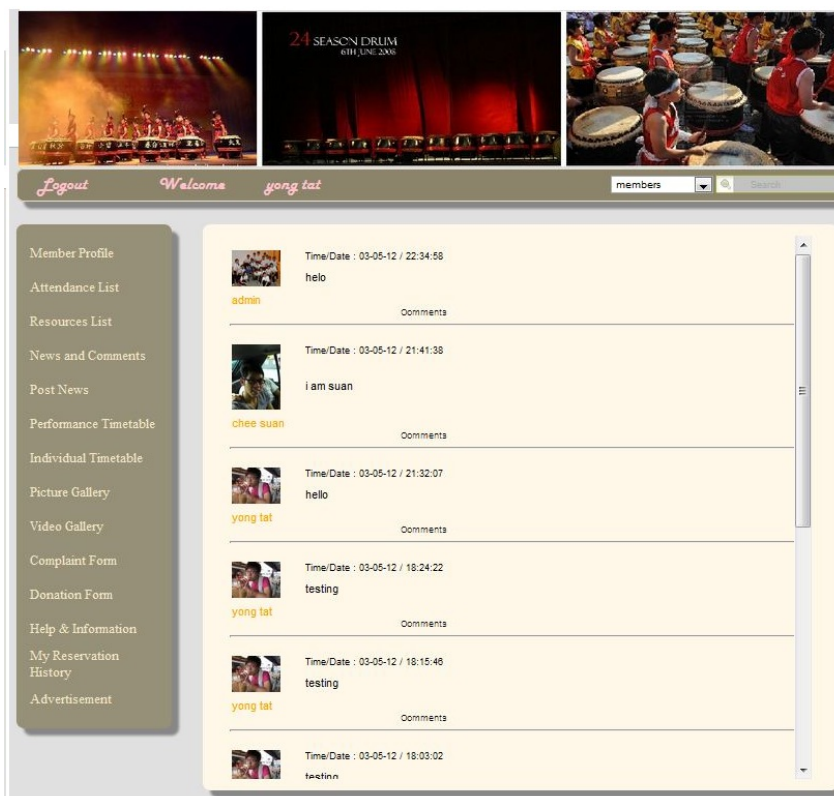


Figure 4.9: 24SDSMS New and Comment – Member

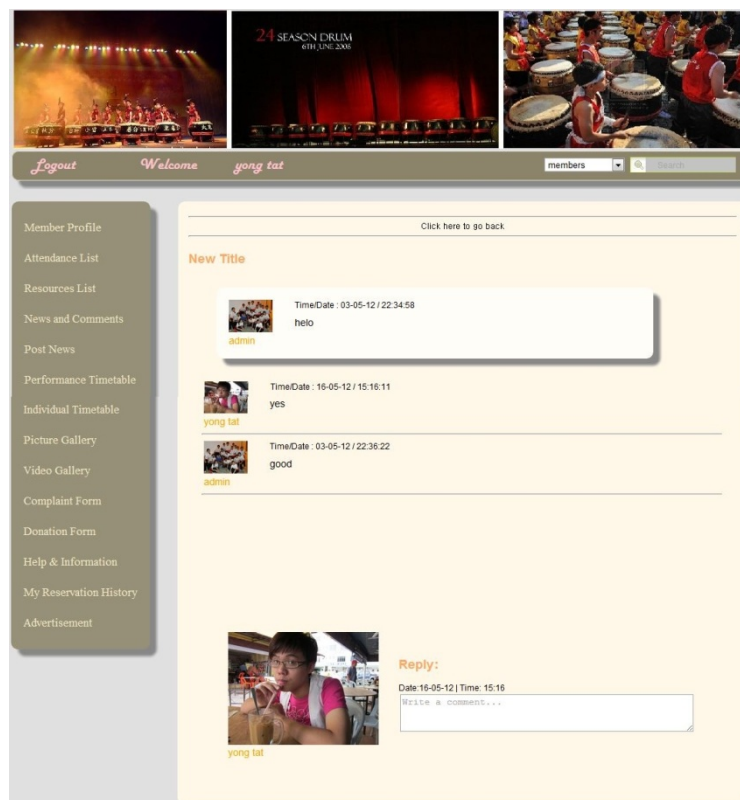


Figure 4.10: 24SDSMS New and Comment1 – Member

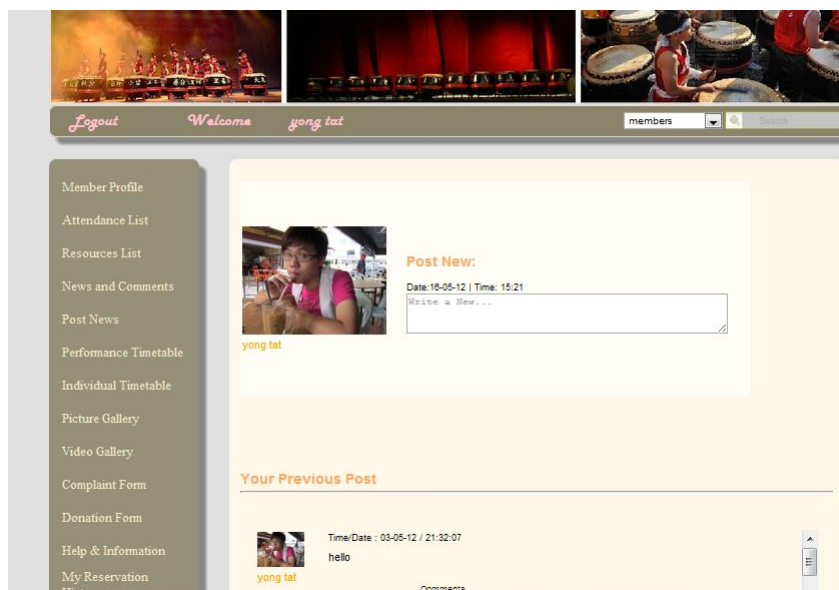


Figure 4.11: 24SDSMS New and Comment2 – Member

4.3.6 Performance List

This module provides for member to view all performance with performance date, venue and time as shown in Figure 4.12. Member can view the list of performers in the selected thread as shown in Figure 4.13. The coding implementation for add new topic function is shown in Appendix B9 and add new reply function is shown in Appendix B10.

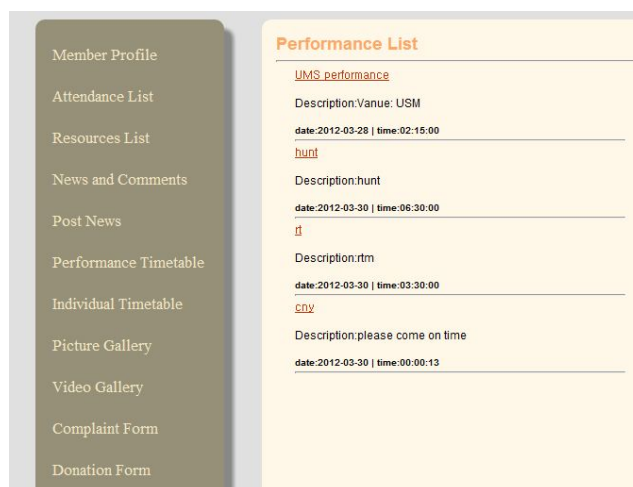


Figure 4.12: 24SDSMS Performance Timetable– Member: coding

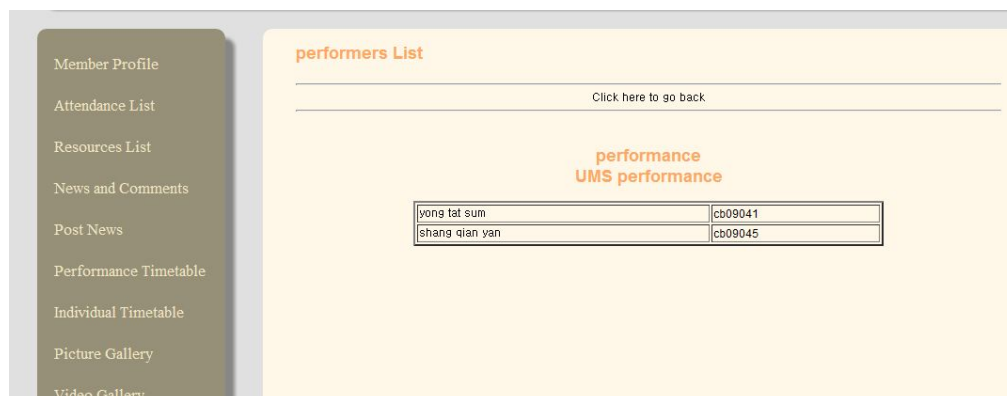


Figure 4.13: 24SDSMS Performers Name List – Member: coding

4.3.7 Individual Timetable

This module provides for member to view individual performance with performance date, venue and time as shown in Figure 4.14. The performances are category in 3 types which is today, upcoming and all performance. Member can view the list of performers in the selected thread as shown in Figure 4.15. The coding implementation for add new topic function is shown in Appendix B11 and add new reply function is shown in Appendix B12.

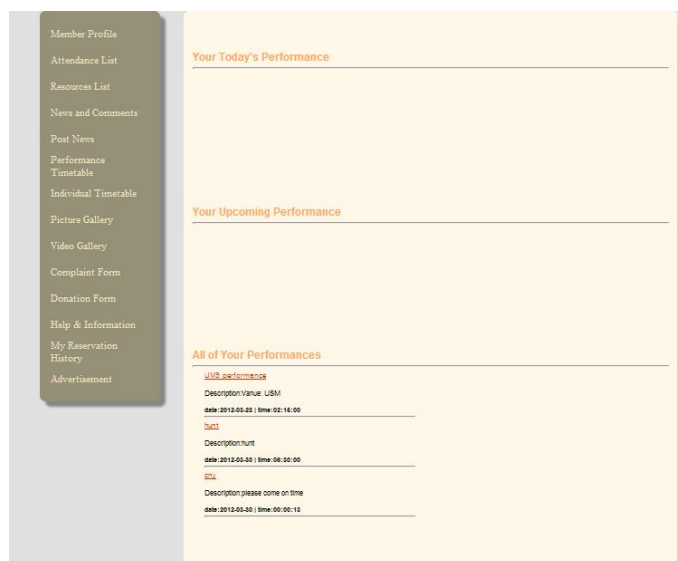


Figure 4.14: 24SDSMS Individual Timetable– Member

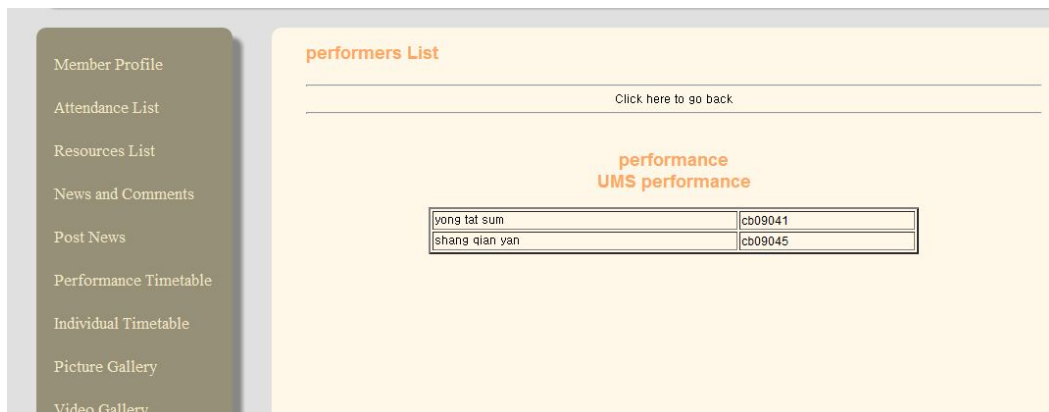


Figure 4.15: 24SDSMS Performers Name List – Member

4.3.8 Pictures

This module allow member to view pictures according to album as show in Figure 4.16 and 4.17 and coding to show the albums and picture show in Appendix B13 and Appendix B14.

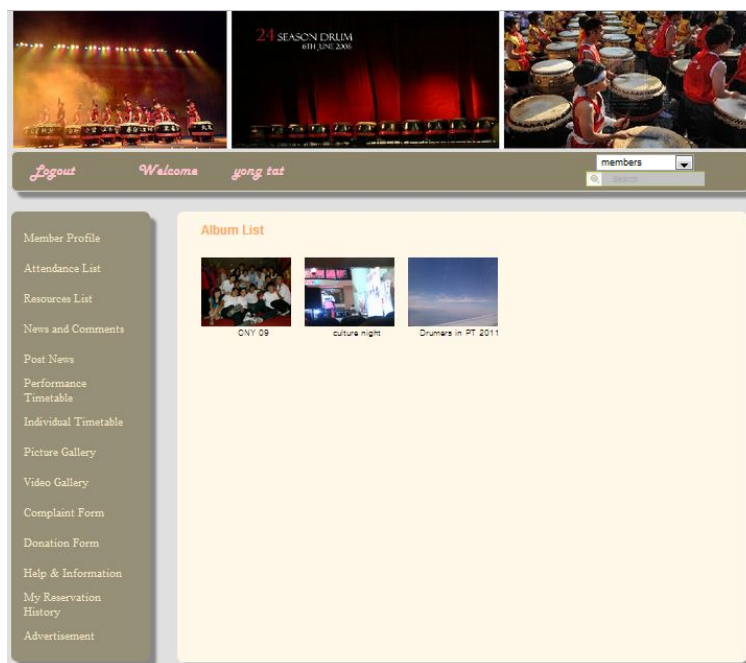


Figure 4.16: 24SDSMS Albums– Member

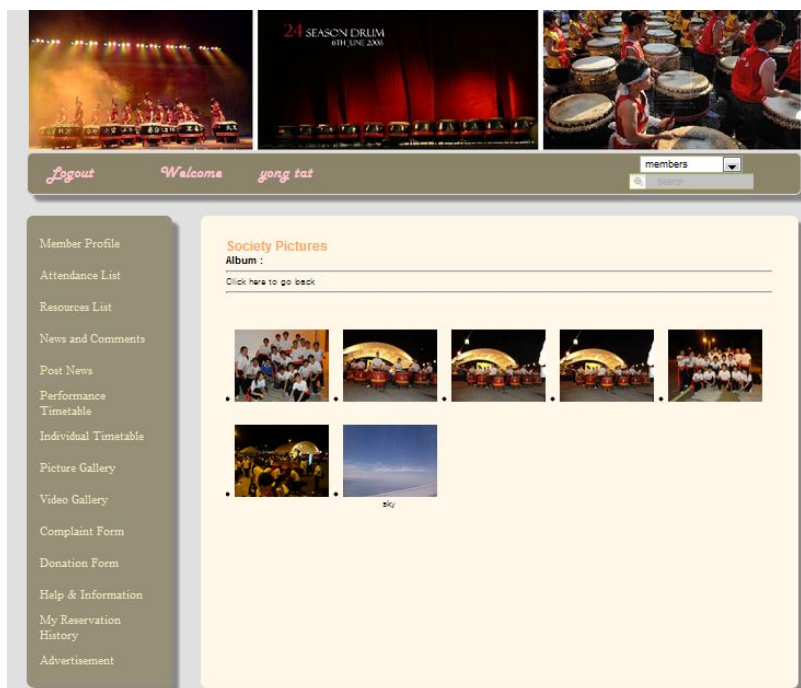


Figure 4.17: 24SDSMS Pictures– Member

4.3.9 Complaint

This module allow member to complaint and give some suggestion. Category of complaint can be choose as show in figure 4.18 and coding show in Appendix B15

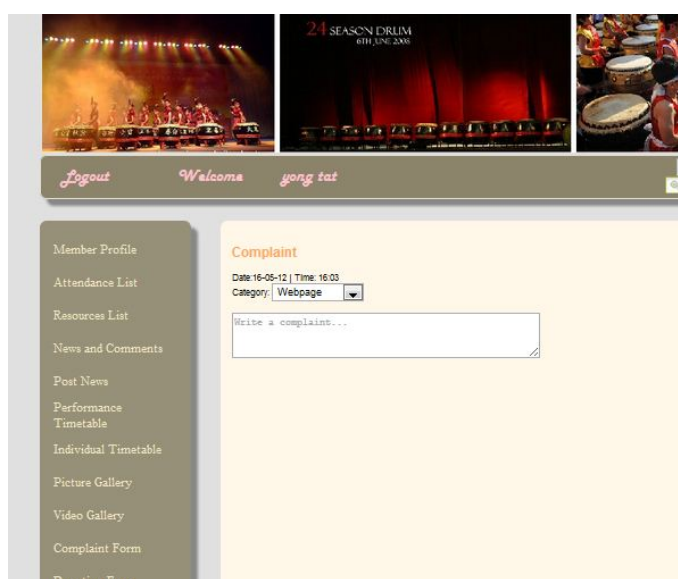


Figure 4.18: 24SDSMS complaint – Member

4.3.10 Help and Information

This module control by administrator, member can view the help and information about system as shows in Figure 4.19

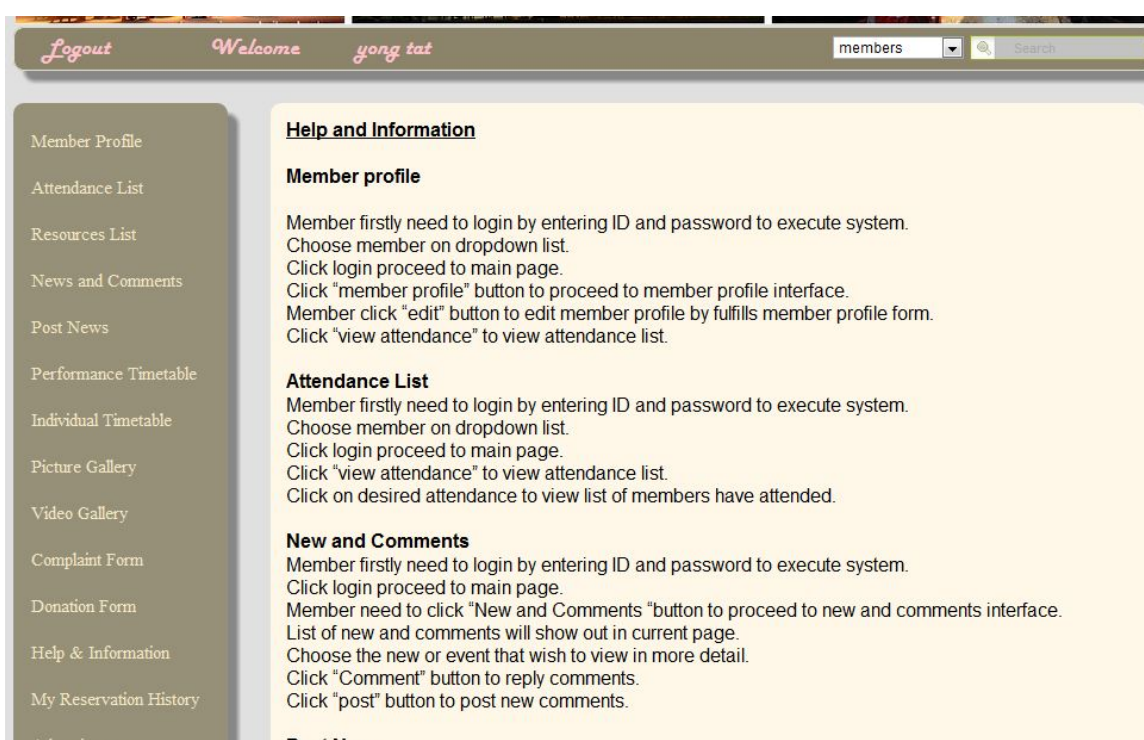


Figure 4.19: 24SDSMS help and information – Member: Coding

4.3.11 Reservation History

This module can let member to view their reserved history as show in Figure 4.20.

Confirmation Code	41
Reserved Period	2012-05-15 - 2012-05-16
Name	yong tat
Matric ID	cb09041
Resources	Gloves
Quantity	1
Reserved Date	12-05-15
Reserved Time	20:40:17
Status	pending
Confirmation Code	40
Reserved Period	2012-05-15 - 2012-05-16
Name	yong tat
Matric ID	cb09041
Resources	Gloves
Quantity	1
Reserved Date	12-05-15
Reserved Time	20:39:55
Status	pending
Confirmation Code	39
Reserved Period	2012-05-15 - 2012-05-16
Name	yong tat
Matric ID	cb09041

Figure 4.20: 24SDSMS Reserved History– Member

4.3.12 Advertisement

This Module allow member to view the advertisement that posted by administrator. The list of advertisement as show in Figure 4.21 is given for member to choose. The picture and the description will show out once member select the advertisement list as show in Figure 4.22.

[Click here to go back](#)

Advertisement List

- New photos for CNY 2012 has been uploaded!!!
- Next performance will held on 2nd May 2012, Are you get ready?
- Get a chance to win a i-pod
- Performance video uploaded!!
-
-

Figure 4.21: 24SDSMS Advertisement List– Member



Figure 4.22: 24SDSMS Advertisement Content – Member

4.3.13 Search

Search function is provided for member to search information base on member, pictures, video and my reservation as show in figure 4.23. Coding for search result show in Appendix B20

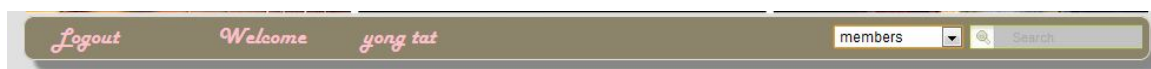


Figure 4.23: 24SDSMS Search Function – Member

4.3.14 Main Page- Administrator

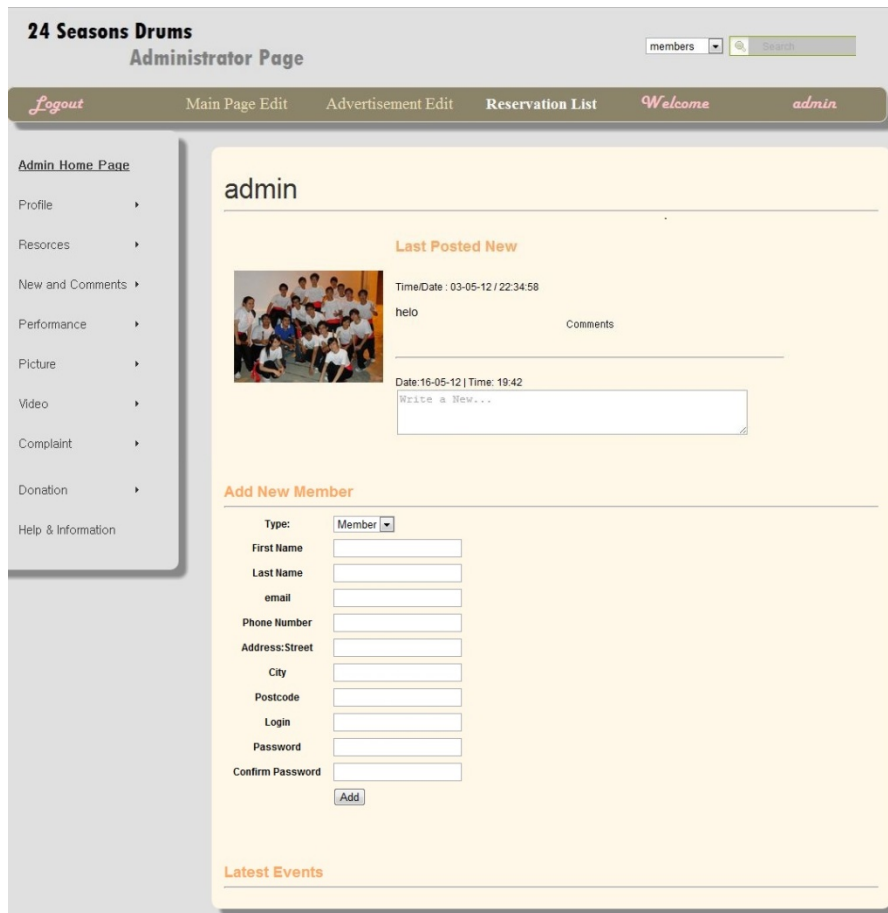





Figure 4.24: 24SDSMS Main Page –Administrator

4.3.15 Main Page Edit

This module allow administrator to edit information on main page, change banner or select menu options to show on member system. Coding for upload banner will show on Appendix B21, coding for sidebar edit will show on Appendix B22, Main content top will show on Appendix B23 and Menu List selection will show on Appendix B24.

Main Page Design

Banner

No file chosen

Sidebar Edit

title

B U [Icons] Font Size Font Family Font Format

The objective of this website are:-

text

B U [Icons] Font Size Font Family Font Format

-To automate the managing process of 24 season drums society.

- To provide a site to let open users or members to get more information about the society.

Main Content-Top

title

B U [Icons] Font Size Font Family

About Our Society

text

B U [Icons] Font Size Font Family

The 24 season drums society, management system is developed to make flow of management can process with more efficiency. The system will provide basic information and background of the society.

If you have any problem please come our help and information page [Click Here](#)

title

B U [Icons] Font Size Font Family

About Our Society

text

B U [Icons] Font Size Font Family

The 24 season drums society, management system is developed to make flow of management can process with more efficiency. The system will provide basic information and background of the society.

If you have any problem please come our help and information page [Click Here](#)

Menu List

Member Profile	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Attendance List	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Resources List	<input checked="" type="radio"/> show	<input type="radio"/> Hide
News and Comments	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Post News	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Performance Timetable	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Individual Timetable	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Picture Gallery	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Video Gallery	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Complaint Form	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Donation Form	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Help & information	<input checked="" type="radio"/> show	<input type="radio"/> Hide
My Reservation History	<input checked="" type="radio"/> show	<input type="radio"/> Hide
Advertisement	<input checked="" type="radio"/> show	<input type="radio"/> Hide

Figure 4.25: 24SDSMS Main Page Edit –Administrator

4.3.16 Advertisement Management

This module provides the capability for admin to manage advertisement as shown in Figure 4.26. All advertisement listed as shown in Figure 4.26. Admin can add new advertisement by enter the advertisement title, content and also upload a

new advertisement poster. Coding for upload poster and store data show in Figure Appendix B25

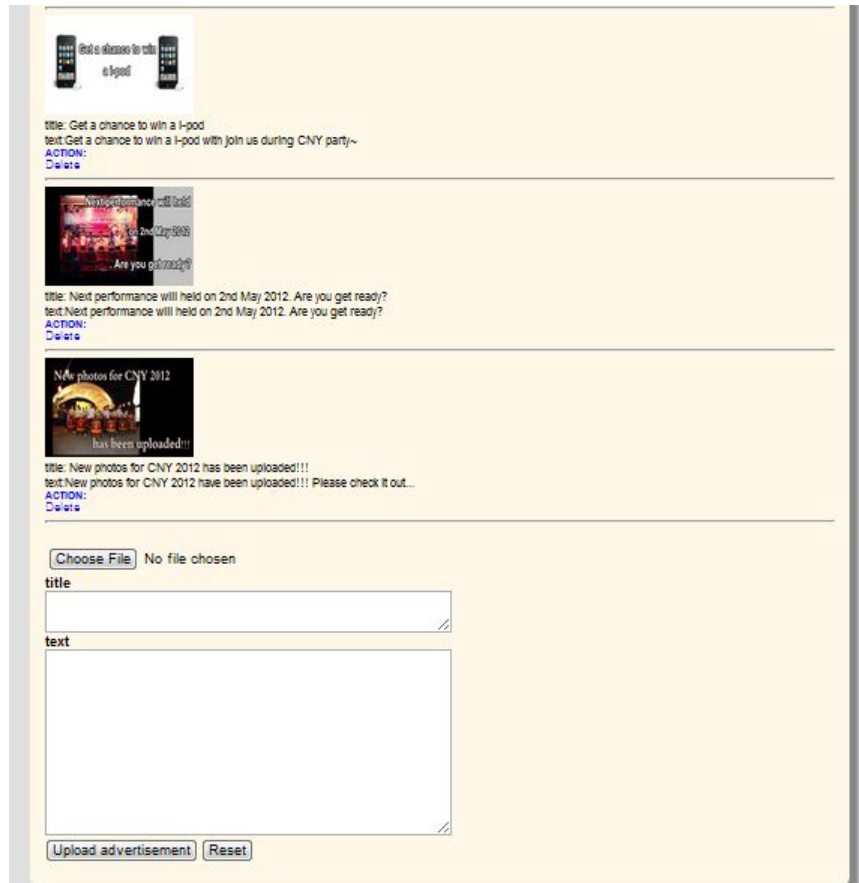


Figure 4.26: 24SDSMS Advertisement Manager –Administrator

4.3.17 Reservation Management

This module provides the capability for admin to manage the reservation as shown in Figure 4.27. All resources and reservation status listed as shown in Figure 4.28. All reservation display according to reservation status as show in Figure 4.29.

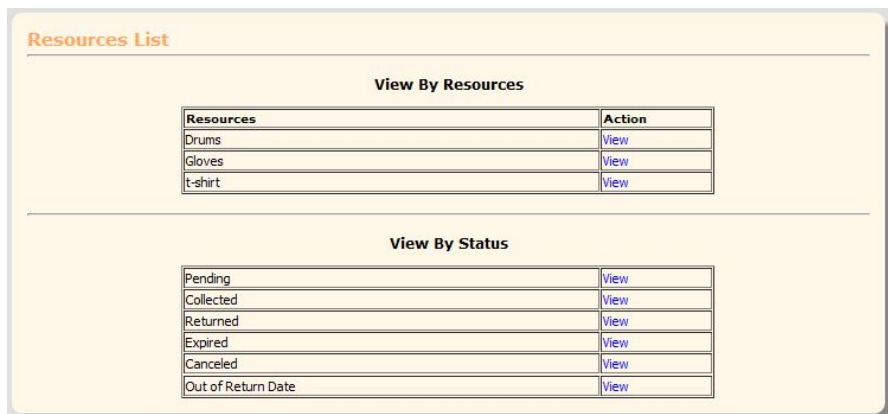


Figure 4.27: 24SDSMS Reservation Manager –Administrator

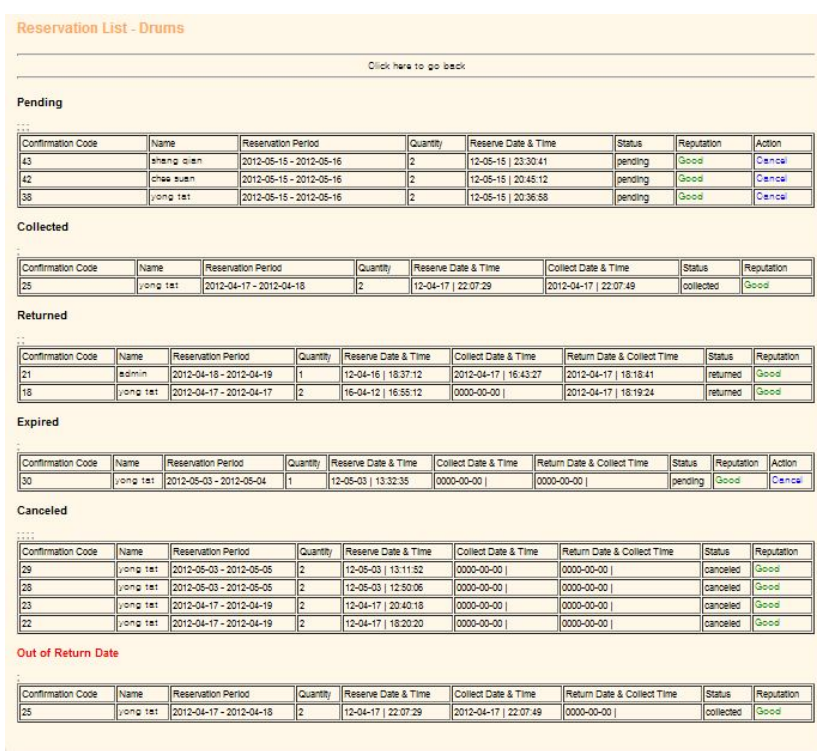


Figure 4.28: 24SDSMS Reservation Manager –Administrator

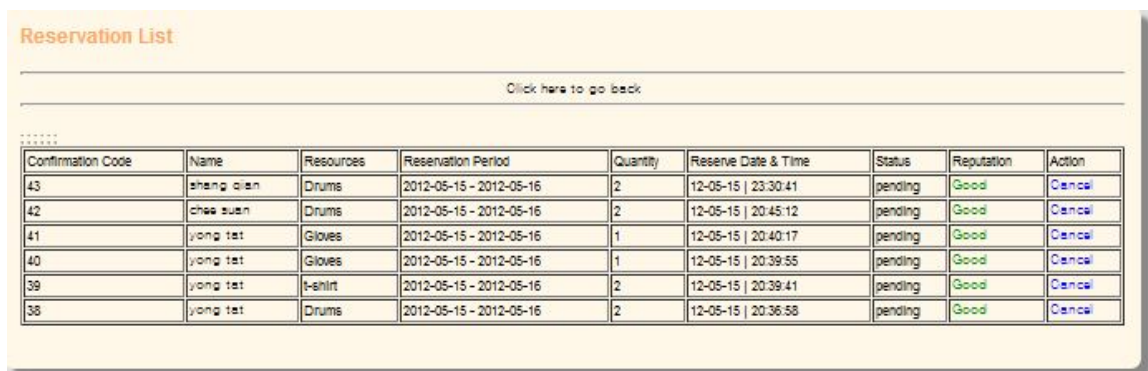


Figure 4.29: 24SDSMS Reservation Manager –Administrator

4.3.18 Profile Management

This module provides the capability for admin to manage the member profile as shown in Figure 4.30. Admin can edit, delete or add the new member as shown in Figure 4.30. Admin can edit the profile of member include change their profile picture as show in Figure 4.31. The coding implementation for add new member function is shown in Appendix B27. The coding for update member profile is show in Appendix B28.

Members List

yong tat, sum	Action: Edit Delete
onee suan, lu	Action: Edit Delete
ed, ded	Action: Edit Delete
admin, admin	Action: Edit Delete
amy, tan	Action: Edit Delete
shang qian, yan	Action: Edit Delete

New Member

Type:

First Name:

Last Name:

email:

Phone Number:

Address: Street:

City:

Postcode:

Login:


Password:

Confirm Password:

Figure 4.30: 24SDSMS Member Profile Manager –Administrator

Profile Picture

Click here to go back



No file chosen

Profile Information

First Name	<input type="text" value="yong tat"/>
Last Name	<input type="text" value="sum"/>
email	<input type="text" value="kono_yakuza@hotmail"/>
Phone Number	<input type="text" value="016-3917794"/>
Address: Street	<input type="text" value="no2, jalan cempaka"/>
City	<input type="text" value="ampang"/>
Postcode	<input type="text" value="68000"/>
Login	<input type="text" value="cb09041"/>
Reputation	<input checked="" type="radio"/> Good <input type="radio"/> Banned
Password	<input type="text"/>
Confirm Password	<input type="text"/>

Figure 4.31: 24SDSMS Member Profile Manager –Administrator

4.3.19 Attendance Management

This module provides the capability for admin to manage the attendance as shown in Figure 4.32. Admin can add the new event as shown in Figure 4.32. The metrics number key in by attended member as show in Figure 4.33. The coding implementation for add new event and key in metrics number is shown in Appendix B29.

Attendance List

IN
date: 2012-03-24 | time:
VIEW / EDIT

training
date: 2012-03-24 | time:
VIEW / EDIT

training 2
date: 2012-03-28 | time:
VIEW / EDIT

Seminar
date: 2012-03-28 | time: 12:36pm
VIEW / EDIT

New Attendance

Date : 18 | May | 2012

Time:

Event / Training:

Figure 4.32: 24SDSMS Attendance Management –Administrator

[Click here to go back](#)

Event Title:
training

2012-03-24 |

Please Key in your Matrics Number

Figure 4.33: 24SDSMS Attendance Management –Administrator

4.3.20 Resources Management

This module provides the capability for admin to manage the resources management as shown in Figure 4.34. Member also can correct or return resources by key in confirmation code as show in Figure 4.35 and Figure 4.36. Admin can view all the reservation as shown in Figure 4.37. The confirmation of correct and return resources receipt will be display as show in Figure 4.38 and Figure 4.39. The coding implementation for add, update and delete function for resources is shown in

Appendix B30, Appendix B31 and Appendix B3. Coding implementation for collect and return resources is show in Appendix B33 and Appendix B34.

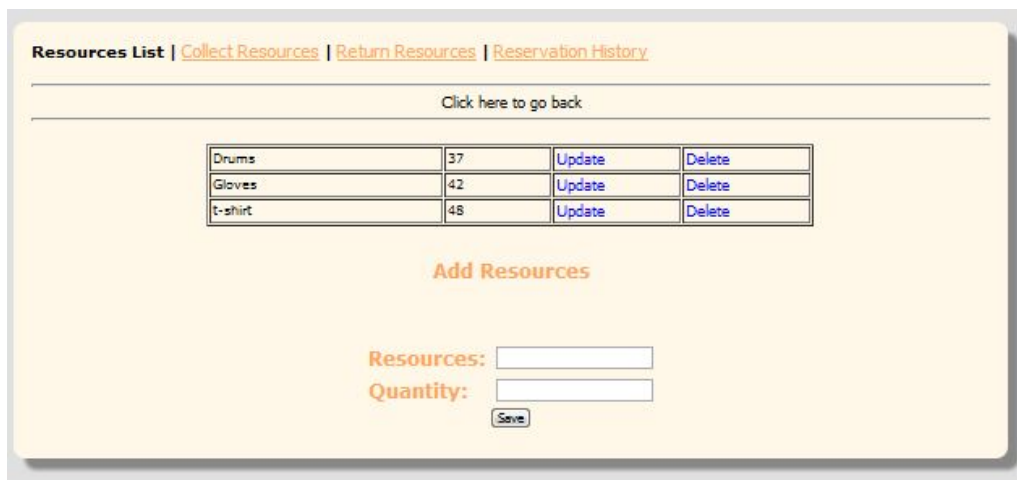


Figure 4.34: 24SDSMS Resources Management –Administrator

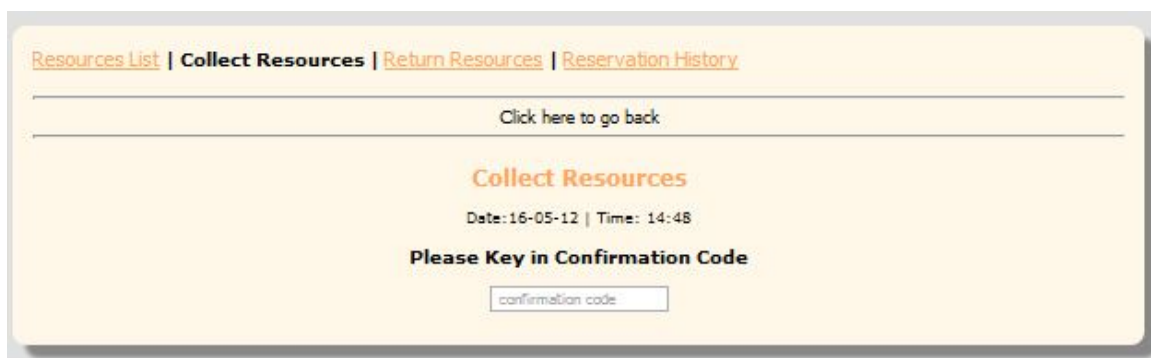


Figure 4.35: 24SDSMS Collect Resources –Administrator

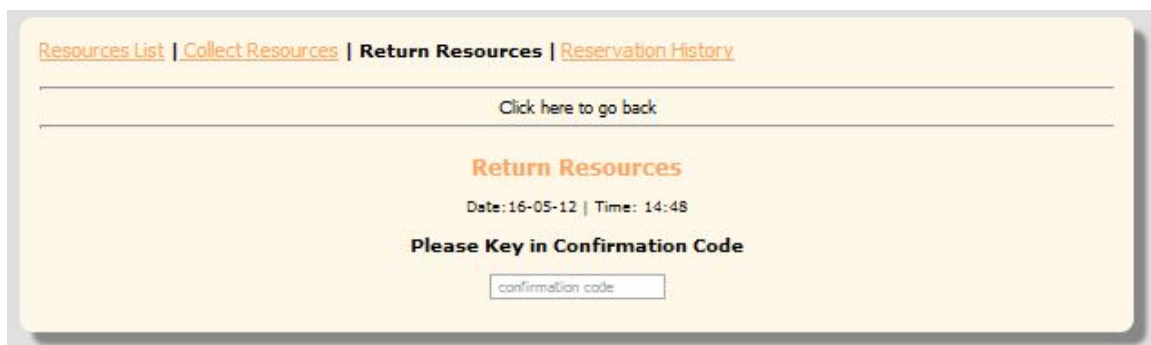


Figure 4.36: 24SDSMS Return Resources –Administrator

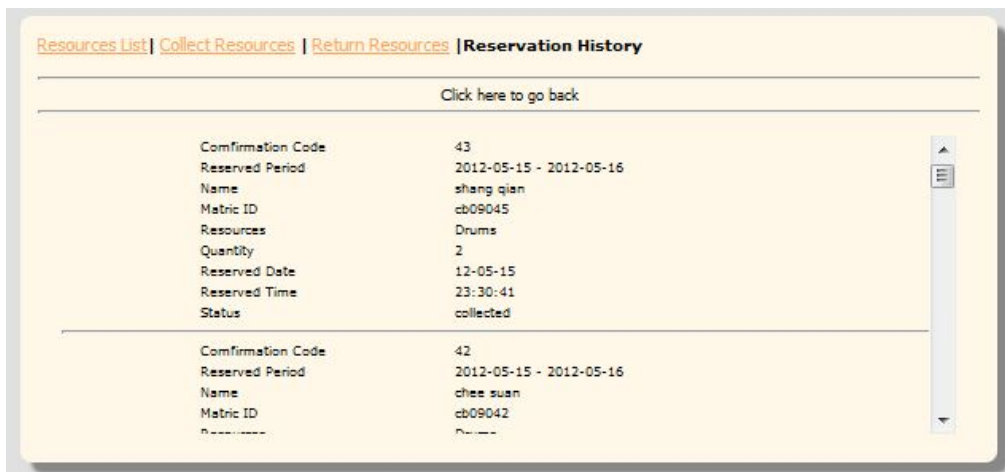


Figure 4.37: 24SDSMS Reservation History –Administrator



Figure 4.38: 24SDSMS Collect Resources Receipt –Administrator

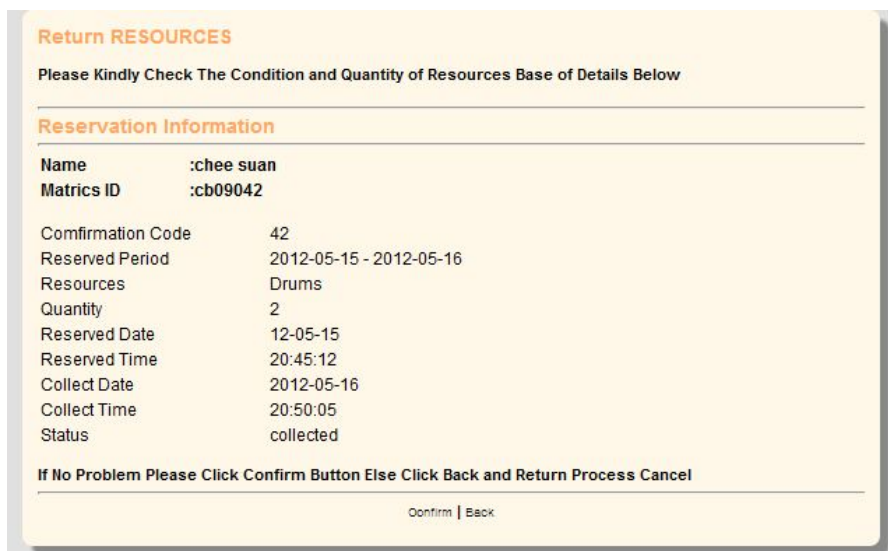
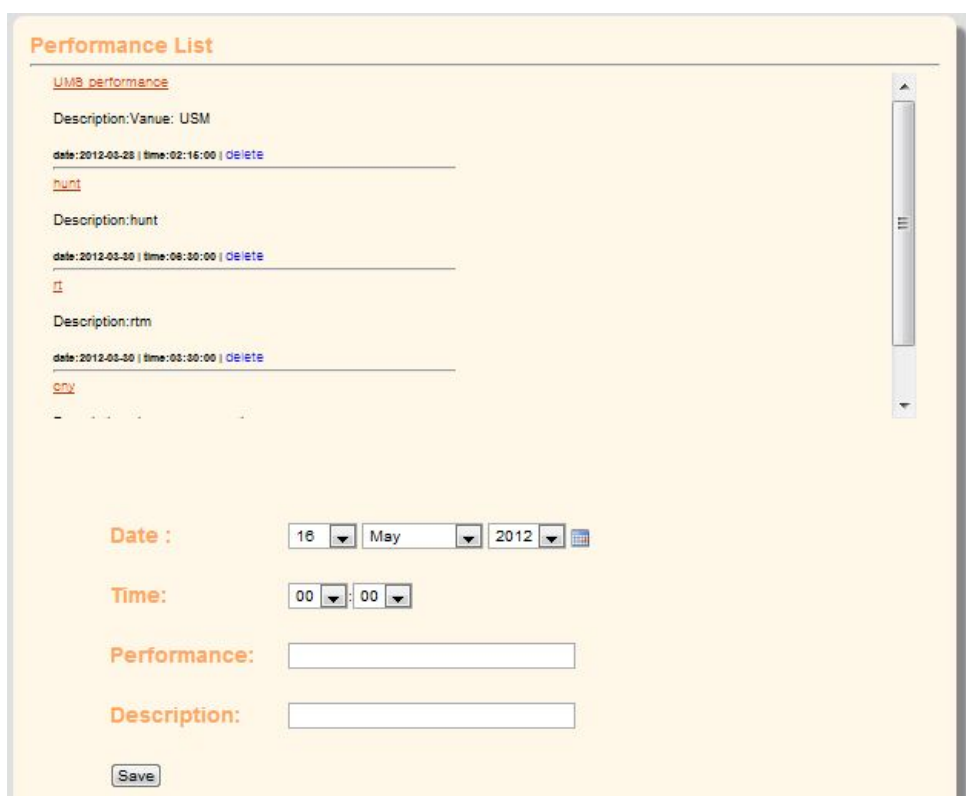


Figure 4.39: 24SDSMS Return Resources Receipt –Administrator

4.3.21 Performance Management

This module provides the capability for admin to manage the performance as shown in Figure 4.40. The admin can key in the performers by insert metrics number as show in Figure 4.41. The coding implementation for add new performance function and add new performers is shown in Appendix B35 and Appendix B36.



The screenshot displays a web interface titled "Performance List". It features a scrollable list of performance entries. Each entry includes a description, a date, and a time, along with a "Delete" link. Below the list, there are input fields for "Date", "Time", "Performance", and "Description", and a "Save" button.

Performance List

UMB performance
Description: Vanue: USM
date: 2012-03-28 | time: 02:16:00 | [Delete](#)

hunt
Description: hunt
date: 2012-03-30 | time: 08:30:00 | [Delete](#)

rtm
Description: rtm
date: 2012-03-30 | time: 03:30:00 | [Delete](#)

ony

Date : 16 May 2012

Time: 00:00

Performance:

Description:

Figure 4.40: 24SDSMS Performance Management–Administrator

Figure 4.41: 24SDSMS add Performers–Administrator

4.3.22 Pictures Management

This module provides the capability for admin to manage the pictures as shown in Figure 4.42 and Figure 4.43. The admin can create new album by insert album name as show in Figure 4.42. Admin can upload pictures in created album as show in Figure 4.43. The coding implementation for create new album function and add new pictures is shown in Appendix B37 and Appendix B38.

Figure 4.42: 24SDSMS Create album–Administrator

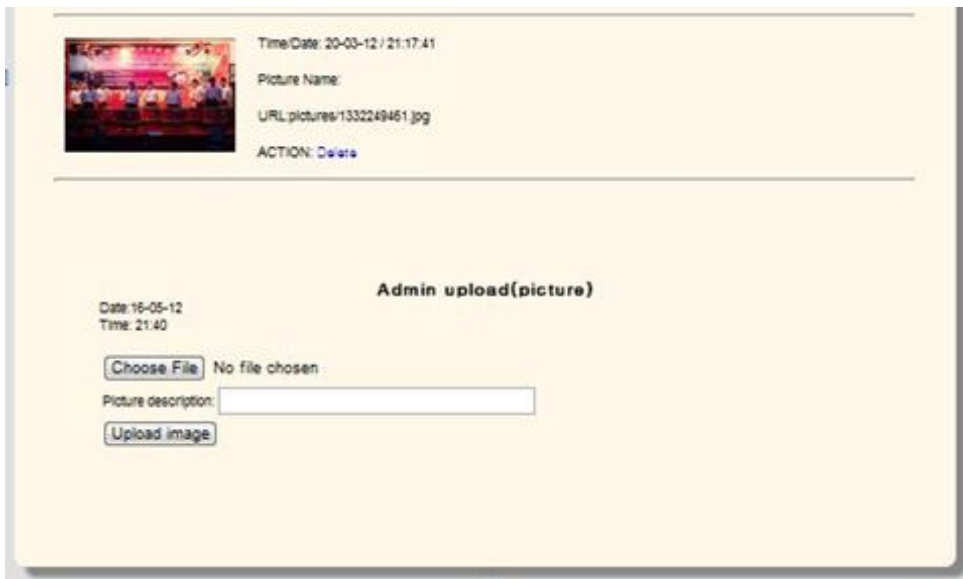


Figure 4.43: 24SDSMS Add Picture–Administrator

4.3.23 Complaint Management

This module provides the capability for admin to manage the complaint sent by member and open user as shown in Figure 4.44. The coding implementation for view complaint list is shown in Appendix B39.

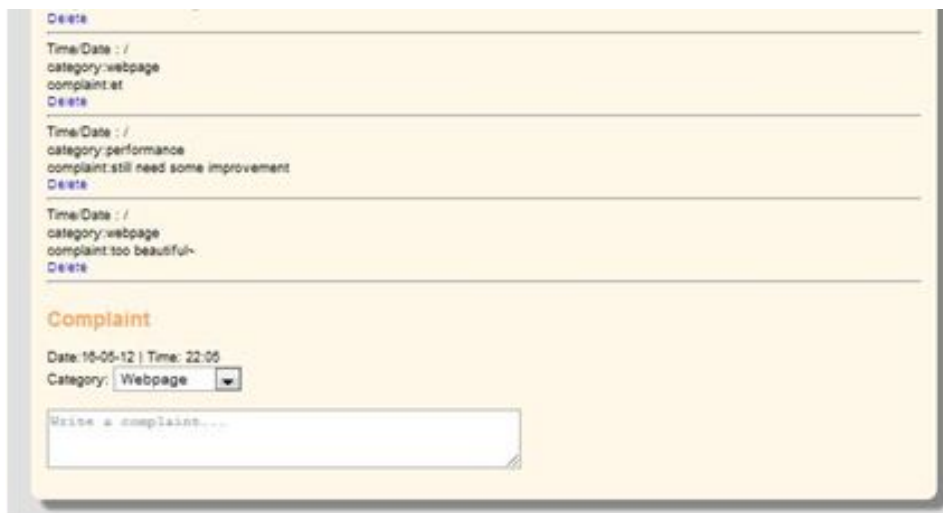


Figure 4.44: 24SDSMS Complaint Management –Administrator

CHAPTER 5

RESULT AND DISCUSSION

This chapter will be devoted to the testing result of the system and discussion on the result is presented.

5.1 User Acceptance Testing

User acceptance testing (UAT) is a formal testing conducted to determine whether a completed system satisfies its acceptance criteria and to enable the user to determine whether to accept the system or not. The acceptance criteria are based on following:

- i) Major functionality
- ii) Interface appearance
- iii) Different level of access
- iv) Performance
- v) Capacity
- vi) Accuracy
- vii) Reliability
- viii) Security
- ix) Usability
- x) Exception handling

5.1.1 Software Tester

The 24 Season Drums Society Management System (24 SDSMS) is tested using UAT. The staffs of 24 Season Drums are the tester for the system. The details of the tester are shown in table 5.1.

Table 5.1: Tester Details

No.	Tester	Initials	Position	Role
1	Lu Chee Suan	LCS	Member of Society	Act as potential member
2	Cheah Phooi Keat	TKT	UMP student	Act as potential open user
3	Yan Shang Qian	YSQ	President of Society	Act as Administrator to manage the system

5.1.2 User Acceptance Testing Result

The UAT is categorized to seven acceptance criteria as mention above to test out. Testing action is design to evaluate the criteria of the 24SDSMS. Each of them exercises a particular operating condition of the user's environment or features of the system, and will result in a pass or fail. The details of the UAT action and result on 24SDSMS are shown in Appendix B. The sign off and acknowledgment of the society and open user is shown in Appendix C. The summary of the result of UAT on 24SDSMS is shown in table 5.2.

Table 5.2: Summary of User Acceptance Testing Result

No.	Tester	Acceptance Criteria	No. of Action	No. of Pass	No. of Fail
1	Lu Chee Suan (LCS)	Major Functionality	8	8	0
		Interface Appearance	4	4	0
		Different Level of Access	4	4	0
		Accuracy	8	8	0
		Capacity	2	2	0
2	Cheah Phooi Keat (TKT)	Performance	2	2	0
		Reliability	2	2	0
3	Yan Shang Qian (YSQ)	Security	2	2	0
		Usability	4	4	0
		Exception Handling	4	4	0
	Total		40	40	0

5.2 System Testing

System testing is the process of performing a variety of test on a system to explore functionality or to identify problems. Testing Tool (Oracle Application Testing Suits) will be use to test the functionality of the system to determine whether a completed system satisfies its functionality. The acceptance criteria are base on following

- I. Exception handling
- II. Security
- III. Capacity
- IV. Usability

5.2.1 System Tester

The 24 Season Drums Society Management System (24 SDSMS) is tested using Testing Tool (Oracle Application Testing Suits). The student of Software Engineering is the tester for the system. The details of the tester are shown in table 5.3.

Table 5.3: Tester Details

No.	Tester	Initials	Position	Role
1	Ang Tzy Guan	ATG	student of Software Engineering	Act as software tester to test 24SDSMS

5.2.2 Testing Modules

The 24 Season Drums Society Management System (24 SDSMS) is tested using Testing Tool (Oracle Application Testing Suits) which according to the functions of the system. The functions which will be test are shown in following:

- I. User Login
- II. Search Function
- III. Post new and reply
- IV. Resources reservation

5.2.2.1 User Login

This function allow user to login as member or administrator. The member and administrator can only access when password and login id correctly key in, else will proceed to login fail page to warn user. Testing result show in Figure 5.1

Section	Name	Duration (sec)	Result
[-] Initialize	Initialize Total (sec)	0.770	Passed
	Launch Browser	0.647	Passed
[-] Iteration1	Iteration Total (sec)	27.171	Passed
	[-] [1] No Title	0.734	Passed
	Click button("@name=Submit")	0.718	Passed
	[-] [2] Login Failed (/login-failed.php)	9.839	Passed
	WaitForPage	0.597	Passed
	Think: 2.903 secs.	2.905	Passed
	Click textBox("@id=login")	0.317	Passed
	Think: 1.534 secs.	1.536	Passed
	SetText textBox("@id=login") cb09041	0.127	Passed
	Think: 0.673 secs.	0.674	Passed
	PressTab textBox("@id=login")	0.879	Passed
	Think: 1.244 secs.	1.244	Passed
	SetPassword textBox("@id=password") *****	0.423	Passed
	Think: 0.925 secs.	0.927	Passed
	PressEnter textBox("@id=password")	0.127	Passed
	[-] [3] 24 Season Drums-main page (/login.php)	3.009	Passed
	WaitForPage	2.101	Passed
	Think: 0.446 secs.	0.447	Passed
	Click link("@text=Logout")	0.435	Passed
	[-] [4] Logged Out (/logout.php)	7.041	Passed
	WaitForPage	0.539	Passed
	Think: 3.151 secs.	3.152	Passed
	Click textBox("@id=login")	0.299	Passed
	Think: 0.92 secs.	0.922	Passed
	SetText textBox("@id=login") admin	0.108	Passed
	Think: 0.055 secs.	0.055	Passed
	PressTab textBox("@id=login")	0.126	Passed
	Think: 0.041 secs.	0.042	Passed
	SetPassword textBox("@id=password") *****	0.095	Passed
	Think: 0.29 secs.	0.291	Passed
	PressTab textBox("@id=password")	0.116	Passed
	Think: 0.782 secs.	0.783	Passed
	SelectOptionByText selectBox("@name=admin") admin	0.116	Passed
	Think: 0.025 secs.	0.026	Passed
	PressTab element("@name=admin")	0.138	Passed
	Think: 0.064 secs.	0.065	Passed
	PressEnter button("@name=Submit")	0.124	Passed
	[-] [5] 24 Season Drums-main page (/admin.php)	4.369	Passed
	WaitForPage	0.883	Passed
	Think: 3.11 secs.	3.111	Passed
	Click link("@text=Logout")	0.363	Passed
	[-] [6] Logged Out (/logout.php)	0.487	Passed
	WaitForPage	0.484	Passed
[-] Finish	Finish Total (sec)	0.095	Passed
	Script Total (sec)	28.247	Passed

Test Results Summary

Total	Result	0%	100%
0	Failed		
0	Warning		
16	Passed		

Figure 5.1: Testing Result for user login function

From the figure 5.1 shows that there are 16 actions recorded and test and all were pass which include:

- I. Wrong password and login id
- II. Page of member will proceed once enter correct password and login id.
- III. Page of administrator will proceed once enter correct password and login id.
- IV. Logout

5.2.2.2 Search Function

This function allow user to search information base on member, pictures, video, and reservation. For search of member, user can search by key in name or metrics id of member. Pictures and video base on name and reservation can search by resources name and confirmation code.

Script Summary

Section	Name	Duration (sec)	Result
[-] Initialize	Initialize Total (sec)	1.068	Passed
	Launch Browser	0.672	Passed
[-] Iteration1	Iteration Total (sec)	48.587	Passed
	[-] [7] 24 Season Drums-picture (/search.php)	21.282	Passed
	WaitForPage	15.689	Passed
	Think: 3.438 secs.	3.442	Passed
	Click textBox("@name=name")	0.468	Passed
	Think: 0.735 secs.	0.736	Passed
	SetText textBox("@name=name") cb09041	0.109	Passed
	Think: 0.287 secs.	0.289	Passed
	PressEnter textBox("@name=name")	0.361	Passed
	[-] [8] 24 Season Drums-picture (/search.php)	11.067	Passed
	WaitForPage	2.089	Passed
	Think: 4.533 secs.	4.534	Passed
	Click textBox("@name=name")	0.304	Passed
	Think: 2.254 secs.	2.255	Passed
	SelectOptionByText selectBox("@name=category") Pictures	0.102	Passed
	Think: 0.045 secs.	0.046	Passed
	Click textBox("@name=name")	0.286	Passed
	Think: 0.705 secs.	0.706	Passed
	SetText textBox("@name=name") veuve	0.140	Passed
	Think: 0.426 secs.	0.427	Passed
	PressEnter textBox("@name=name")	0.115	Passed
	[-] [9] 24 Season Drums-picture (/search.php)	6.296	Passed
	WaitForPage	0.786	Passed
	Think: 2.877 secs.	2.879	Passed
	SelectOptionByText selectBox("@name=category") My Reservation	0.119	Passed
	Think: 0.572 secs.	0.573	Passed
	Click textBox("@name=name")	0.292	Passed
	Think: 1.135 secs.	1.137	Passed
	SetText textBox("@name=name") pending	0.097	Passed
	Think: 0.279 secs.	0.280	Passed
	PressEnter textBox("@name=name")	0.110	Passed
	[-] [10] 24 Season Drums-picture (/search.php)	8.297	Passed
	WaitForPage	0.871	Passed
	Think: 5.421 secs.	5.422	Passed
	SelectOptionByText selectBox("@name=category") My Reservation	0.211	Passed
	Think: 0.066 secs.	0.067	Passed
	Click textBox("@name=name")	0.287	Passed
	Think: 0.839 secs.	0.840	Passed
	SetText textBox("@name=name") 30	0.100	Passed
	Think: 0.341 secs.	0.341	Passed
	PressEnter textBox("@name=name")	0.115	Passed
	[-] [11] 24 Season Drums-picture (/search.php)	0.670	Passed
	WaitForPage	0.666	Passed
[-] Finish	Finish Total (sec)	0.081	Passed
	Script Total (sec)	50.890	Passed

Test Results Summary

Total	Result	0%	100%
0	Failed	<div style="width: 0%; height: 10px; background-color: red;"></div>	
0	Warning	<div style="width: 0%; height: 10px; background-color: orange;"></div>	
16	Passed	<div style="width: 100%; height: 10px; background-color: green;"></div>	

Figure 5.2: Testing Result for user login function

5.2.2.3 Post New and Reply

This function allow user to post new and also reply for the posted new. Tester test by key in word “system testing” to post a new. After that reply the new with words “testing success”. The Script summary show in Figure 5.3.

Script Summary

Section	Name	Duration (sec)	Result	Summary
Initialize	Initialize Total (sec)	0.805	Passed	
	Launch Browser	0.680	Passed	
Iteration1	Iteration Total (sec)	36.652	Passed	
	[1] 24 Season Drum-News & Comments (P_N.php)	6.865	Passed	
	WaitForPage	1.813	Passed	Screen_Shot.html
	Think: 3.743 secs.	3.745	Passed	
	SetText testArea("@name=10w") system testing	1.284	Passed	
	[1] 24 Season Drum-News & Comments (P_N.php)	14.275	Passed	
	WaitForPage	9.934	Passed	Screen_Shot.html
	Think: 3.329 secs.	3.330	Passed	
	Click link("@name=Comments")	0.736	Passed	
	Text Matching Test: check new, system testing	0.271	Passed	
	[4] 24 Season Drum-News & Comments (P_N2.php)	4.496	Passed	
	WaitForPage	0.523	Passed	Screen_Shot.html
	Think: 3.78 secs.	3.781	Passed	
	SetText testArea("@name=10comment") testing success	0.171	Passed	
	[1] 24 Season Drum-News & Comments (P_N2.php)	10.171	Passed	
	WaitForPage	9.738	Passed	Screen_Shot.html
	Text Matching Test: reply testing, testing success	0.085	Passed	
Finish	Finish Total (sec)	0.000	Passed	
	Script Total (sec)	37.688	Passed	

Test Results Summary

Total	Result	0%	100%
0	Failed		
0	Warning		
5	Passed		

Figure 5.3: Testing Result for Post new and reply

5.2.2.4 Resources Reservation

This function allow member to reserve resources which need to obey several condition. System will stop the reservation process once either one of these conditions are not followed. Tester need to test these entire conditions base on all possible situations. The conditions that limit the member reserve the resources are:

- i. Quantity must be key in
- ii. Reservation cannot exceed more than 3 days
- iii. Each member only can reserve 2 quantities of resources
- iv. Date must be key in
- v. Member only can reserve one time per day.

Script Summary

Section	Name	Duration (sec)	Result
Initialize	Initialize Total (sec)	2.539	Passed
	Launch Browser	2.388	Passed
Iteration1	Iteration Total (sec)	170.492	Passed
	[15] 24 Season Drums-Resources List (/ResourcesList.php)	18.538	Passed
	[16] 24 Season Drums-Resources List (/ResourcesList2.php)	27.234	Passed
	WaitForPage	2.635	Passed
	Think: 2.14 secs.	2.141	Passed
	Click element("@text='Select Date'")	19.867	Passed
	Think: 1.87 secs.	1.872	Passed
	Click element("@text='18'")	0.620	Passed
	[17] 24 Season Drums-Resources List (/calendar_form.php)	4.356	Passed
	WaitForPage	0.446	Passed
	Think: 0.345 secs.	0.346	Passed
	Click element("@text='Select Date'")	0.772	Passed
	Think: 2.156 secs.	2.158	Passed
	Click element("@text='24'")	0.606	Passed
	[18] 24 Season Drums-Resources List (/calendar_form.php)	3.870	Passed
	[19] 24 Season Drums-Resources List (/ResourcesList2.php)	14.123	Passed
	WaitForPage	0.887	Passed
	Think: 9.995 secs.	9.997	Passed
	Text Matching Test: quantity validation, *quantity missing	0.351	Passed
	Text Matching Test: Period Checking, Reservation period cannot more than 3 days	0.074	Passed
	Click element("@text='Select Date'")	0.479	Passed
	Think: 1.313 secs.	1.314	Passed
	Click element("@text='18'")	0.614	Passed
	[20] 24 Season Drums-Resources List (/calendar_form.php)	2.659	Passed
	WaitForPage	0.475	Passed
	Think: 0.36 secs.	0.361	Passed
	Click element("@text='Select Date'")	0.388	Passed
	Think: 0.768 secs.	0.769	Passed
	Click element("@text='19'")	0.599	Passed
	[21] 24 Season Drums-Resources List (/calendar_form.php)	7.713	Passed
	WaitForPage	0.501	Passed
	Think: 0.022 secs.	0.023	Passed
	Click textBox("@name='quantity'")	1.093	Passed
	Think: 3.47 secs.	3.471	Passed
	SetText textBox("@name='quantity'") 5	0.467	Passed
	Think: 1.365 secs.	1.366	Passed
	Click button("@name='Submit'")	0.767	Passed
	[22] 24 Season Drums-Resources List (/ResourcesList2.php)	4.729	Passed
	WaitForPage	0.781	Passed
	Think: 1.425 secs.	1.426	Passed
	Text Matching Test: reserve quantity limitation, *Each member only allowed to reserve 2 resources	0.121	Passed
	Click element("@text='Select Date'")	0.326	Passed
	Think: 1.36 secs.	1.361	Passed
	Click element("@text='18'")	0.614	Passed
	[23] 24 Season Drums-Resources List (/calendar_form.php)	2.005	Passed
	WaitForPage	0.521	Passed
	Think: 0.03 secs.	0.048	Passed
	Click element("@text='Select Date'")	0.331	Passed
	Think: 0.48 secs.	0.482	Passed
	Click element("@text='19'")	0.598	Passed
	[24] 24 Season Drums-Resources List (/calendar_form.php)	4.958	Passed
	WaitForPage	0.598	Passed
	Think: 0.015 secs.	0.016	Passed

Figure 5.4: Testing Result for Resources Reservation

[24]	24 Season Drums-Resources List (/calendar_form.php)	4.958	Passed
	WaitForPage	0.598	Passed Screen Shot Html
	Think: 0.015 secs.	0.016	Passed
	Click textBox("@name='quantity'")	0.804	Passed
	Think: 0.888 secs.	0.889	Passed
	SetText textBox("@name='quantity'") 40	0.448	Passed
	Think: 1.326 secs.	1.327	Passed
	Click button("@name='Submit'")	0.775	Passed
[25]	24 Season Drums-Resources List (/ResourcesList2.php)	13.000	Passed
	WaitForPage	0.818	Passed Screen Shot Html
	Think: 12.577 secs.	10.002	Passed
	Click image("@id='tblbtn_data3'")	0.294	Passed
	Think: 1.276 secs.	1.277	Passed
	Click element("@text='18'")	0.574	Passed
[26]	24 Season Drums-Resources List (/calendar_form.php)	2.106	Passed
	WaitForPage	0.449	Passed Screen Shot Html
	Think: 0.015 secs.	0.017	Passed
	Click element("@text='Select Date'")	0.352	Passed
	Think: 0.649 secs.	0.651	Passed
	Click element("@text='19'")	0.622	Passed
[27]	24 Season Drums-Resources List (/calendar_form.php)	4.353	Passed
	WaitForPage	0.456	Passed Screen Shot Html
	Think: 0.027 secs.	0.028	Passed
	Click textBox("@name='quantity'")	0.779	Passed
	Think: 1.168 secs.	1.169	Passed
	SetText textBox("@name='quantity'") 1	0.446	Passed
	Think: 0.68 secs.	0.681	Passed
	Click button("@name='Submit'")	0.776	Passed
[28]	24 Season Drums-RResources List (/ResourcesList2-confirm.php)	36.653	Passed
[29]	24 Season Drums-main page (/ResourcesHistory.php)	6.639	Passed
[30]	24 Season Drums-Resources List (/ResourcesList.php)	2.290	Passed
[31]	24 Season Drums-Resources List (/ResourcesList2.php)	7.115	Passed
	WaitForPage	0.590	Passed Screen Shot Html
	Think: 1.189 secs.	1.190	Passed
	Click textBox("@name='quantity'")	0.932	Passed
	Think: 1.011 secs.	1.011	Passed
	SetText textBox("@name='quantity'") 1	0.497	Passed
	Think: 1.827 secs.	1.827	Passed
	Click button("@name='Submit'")	1.042	Passed
[32]	24 Season Drums-Resources List (/ResourcesList2.php)	1.385	Passed
	WaitForPage	0.952	Passed Screen Shot Html
	Text Matching Test: Date Validation, *Please insert your date	0.138	Passed
	Text Matching Test: each member can reserve one time per day, *Error:Sorry, you can only reserve for one time on each resource per day.	0.063	Passed
Finish	Finish Total (sec)	0.222	Passed
	Script Total (sec)	173.522	Passed

Test Results Summary		
Total	Result	0%
0	Failed	0
0	Warning	0
39	Passed	100%

Figure 5.4: Testing Result for Resources Reservation (cont)

5.2.2.5 Profile Management

This function allow administrator to manage member profile which include add new member delete member and modify member information. Validations checked by leave blank for all textbox during submit and second time fill-in all information but password and confirm password key in by different words.

Script Summary

Section	Name	Duration (sec)	Result	Summary
Initialize	Initialize Total (sec)	0.736	Passed	
	Launch Browser	0.610	Passed	
Iteration1	Iteration Total (sec)	38.601	Passed	
	[1] No Title	0.463	Passed	
	[2] 24 Season Drums-main page (/register-form.php)	2.947	Passed	
	WaitForPage	0.717	Passed	Screen Shot Html
	Think: 1.758 secs.	1.759	Passed	
	Click button("@name=Submit")	0.432	Passed	
	[3] 24 Season Drums-main page (/register-form.php)	33.567	Passed	
	WaitForPage	11.583	Passed	Screen Shot Html
	Think: 2.456 secs.	2.458	Passed	
	Text Matching Test: Password Validation, *Password missing	0.597	Passed	
	Text Matching Test: first name validation, *First name missing	0.055	Passed	
	Text Matching Test: Last name validation, *Last name missing	0.117	Passed	
	Text Matching Test: Email validation, *Email missing	0.050	Passed	
	Text Matching Test: Phone number Validation, *phone number missing	0.038	Passed	
	Text Matching Test: City Validation, *city missing	0.033	Passed	
	Text Matching Test: Postcode Validation, *Postcode missing	0.029	Passed	
	Text Matching Test: Login ID Validation, *Login ID missing	0.028	Passed	
	Text Matching Test: Confirm Password Validation, *Confirm password missing	0.027	Passed	
	Click textBox("@id=fname")	0.456	Passed	
	Think: 1.346 secs.	1.347	Passed	
	SetText textBox("@id=fname") tester	0.269	Passed	
	Think: 0.296 secs.	0.297	Passed	
	PressTab textBox("@id=fname")	0.293	Passed	
	Think: 0.225 secs.	0.230	Passed	
	SetText textBox("@id=fname") ang	0.219	Passed	
	Think: 0.331 secs.	0.332	Passed	
	PressTab textBox("@id=fname")	0.279	Passed	
	Think: 0.308 secs.	0.309	Passed	
	SetText textBox("@id=email") kono_yakuza@hotmail.com	0.254	Passed	
	Think: 0.111 secs.	0.112	Passed	
	PressTab textBox("@id=email")	0.287	Passed	
	Think: 0.517 secs.	0.518	Passed	
	SetText textBox("@id=phone_no") 0163917794	0.227	Passed	
	Think: 0.318 secs.	0.319	Passed	
	PressTab textBox("@id=phone_no")	0.235	Passed	
	Think: 0.541 secs.	0.542	Passed	
	SetText textBox("@id=street") no2 jalan cempaka	0.217	Passed	
	Think: 0.252 secs.	0.253	Passed	
	PressTab textBox("@id=street")	0.236	Passed	
	Think: 0.353 secs.	0.355	Passed	
	SetText textBox("@id=city") ampanng	0.225	Passed	
	Think: 0.28 secs.	0.281	Passed	
	PressTab textBox("@id=city")	0.241	Passed	
	Think: 0.761 secs.	0.762	Passed	
	SetText textBox("@id=postcode") 68000	0.240	Passed	
	Think: 0.084 secs.	0.085	Passed	
	PressTab textBox("@id=postcode")	0.242	Passed	
	Think: 4.14 secs.	4.142	Passed	
	SetText textBox("@id=login") tester	0.223	Passed	
	Think: 0.156 secs.	0.157	Passed	
	PressTab textBox("@id=login")	0.259	Passed	
	Think: 1.341 secs.	1.342	Passed	
	SetPassword textBox("@id=password") *****	0.546	Passed	
	Think: 0.561 secs.	0.562	Passed	
	PressTab textBox("@id=password")	0.191	Passed	
	Think: 0.064 secs.	0.065	Passed	
	SetPassword textBox("@id=cpassword") *****	0.173	Passed	
	Think: 0.26 secs.	0.261	Passed	
	PressEnter textBox("@id=cpassword")	0.208	Passed	
	[4] 24 Season Drums-main page (/register-form.php)	0.766	Passed	
	WaitForPage	0.667	Passed	Screen Shot Html
	Text Matching Test: Password Matching Test, *Passwords do not match	0.028	Passed	
Finish	Finish Total (sec)	0.107	Passed	
	Script Total (sec)	39.662	Passed	

Test Results Summary

Total	Result	0%	100%
0	Failed	█	
0	Warning	█	
33	Passed	█	

Figure 5.5: Testing Result for Profile Management

5.2.2.6 Resources Management

This function allow administrator to management resources by insert new resources. Tester should test the system by make sure there are warning box appear if invalid input exist. Testing result for resources management shows on Figure 5.6

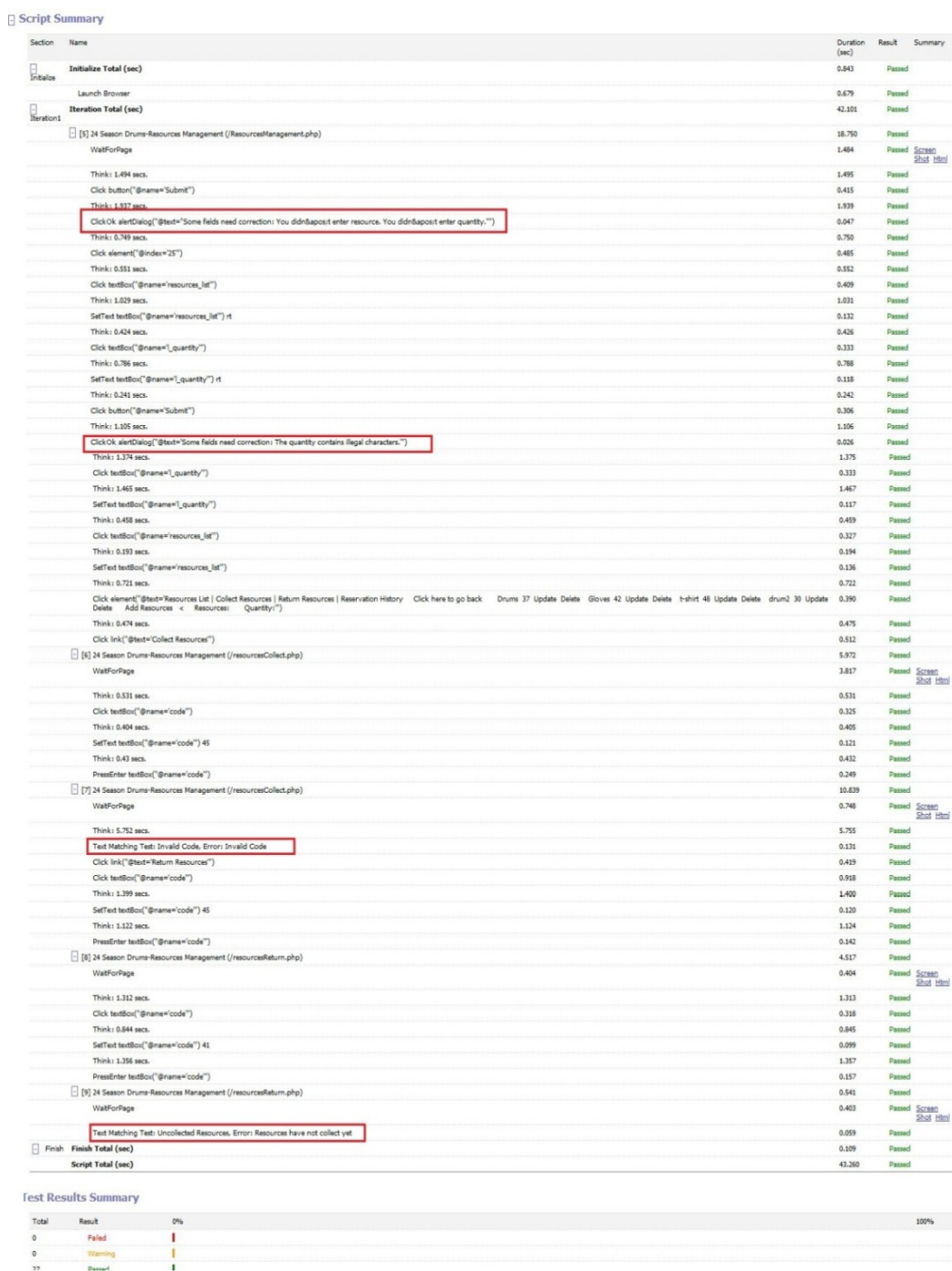


Figure 5.6: Testing Result for Resources Management

5.2.2.7 Reservation Management

This function allow administrator to view all reservation status in table form. Test need to test all table whether match between expected and actual tables.

Script Summary

Section	Name	Duration (sec)	Result	Summary
[-] Initialize	Initialize Total (sec)	1.050	Passed	
	Launch Browser	0.891	Passed	
[-] Iteration1	Iteration Total (sec)	20.601	Passed	
	[10] 24 Season Drums-Reservation Management (ResourcesManagement3.php)	6.677	Passed	
	[11] 24 Season Drums-Reservation Management (ResourcesManagement4.php)	7.414	Passed	
	WaitForPage	0.407	Passed	Screen Shot Html
	Think: 1.689 secs.	1.690	Passed	
	[-] Table Test: Cancel Table	0.875	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.001	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.001	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.000	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.001	Passed	
	Cell 1,6: Expected: 'Collect Date & Time', Actual: 'Collect Date & Time'	0.000	Passed	
	Cell 1,7: Expected: 'Return Date & Collect Time', Actual: 'Return Date & Collect Time'	0.000	Passed	
	Cell 1,8: Expected: 'Status', Actual: 'Status'	0.001	Passed	
	Cell 1,9: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	[-] Table Test: Out of Date Table	0.805	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.001	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.000	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.001	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.000	Passed	
	Cell 1,6: Expected: 'Collect Date & Time', Actual: 'Collect Date & Time'	0.000	Passed	
	Cell 1,7: Expected: 'Return Date & Collect Time', Actual: 'Return Date & Collect Time'	0.000	Passed	
	Cell 1,8: Expected: 'Status', Actual: 'Status'	0.001	Passed	
	Cell 1,9: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	[-] Table Test: Expired Table	0.804	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.001	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.000	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.001	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.000	Passed	
	Cell 1,6: Expected: 'Collect Date & Time', Actual: 'Collect Date & Time'	0.000	Passed	
	Cell 1,7: Expected: 'Return Date & Collect Time', Actual: 'Return Date & Collect Time'	0.000	Passed	
	Cell 1,8: Expected: 'Status', Actual: 'Status'	0.001	Passed	
	Cell 1,9: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	Cell 1,10: Expected: 'Action', Actual: 'Action'	0.002	Passed	
	[-] Table Test: Collected Table	0.766	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.000	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.001	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.000	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.000	Passed	
	Cell 1,6: Expected: 'Collect Date & Time', Actual: 'Collect Date & Time'	0.000	Passed	
	Cell 1,7: Expected: 'Status', Actual: 'Status'	0.001	Passed	
	Cell 1,8: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	[-] Table Test: Pending Table	0.641	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.001	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.000	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.001	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.000	Passed	
	Cell 1,6: Expected: 'Status', Actual: 'Status'	0.000	Passed	
	Cell 1,7: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	Cell 1,8: Expected: 'Action', Actual: 'Action'	0.000	Passed	
	[-] Table Test: Returned Table	0.896	Passed	
	Cell 1,1: Expected: 'Confirmation Code', Actual: 'Confirmation Code'	0.000	Passed	
	Cell 1,2: Expected: 'Name', Actual: 'Name'	0.000	Passed	
	Cell 1,3: Expected: 'Reservation Period', Actual: 'Reservation Period'	0.001	Passed	
	Cell 1,4: Expected: 'Quantity', Actual: 'Quantity'	0.000	Passed	
	Cell 1,5: Expected: 'Reserve Date & Time', Actual: 'Reserve Date & Time'	0.000	Passed	
	Cell 1,6: Expected: 'Collect Date & Time', Actual: 'Collect Date & Time'	0.000	Passed	
	Cell 1,7: Expected: 'Return Date & Collect Time', Actual: 'Return Date & Collect Time'	0.001	Passed	
	Cell 1,8: Expected: 'Status', Actual: 'Status'	0.000	Passed	
	Cell 1,9: Expected: 'Reputation', Actual: 'Reputation'	0.000	Passed	
	Click link("@text=Click here to go back")	0.502	Passed	
	[12] 24 Season Drums-Reservation Management (ResourcesManagement3.php)	1.809	Passed	
	[13] 24 Season Drums-Reservation Management (ResourcesManagement5.php)	2.701	Passed	
	[14] 24 Season Drums-Reservation Management (ResourcesManagement3.php)	0.488	Passed	
[-] Finish	Finish Total (sec)	0.150	Passed	
	Script Total (sec)	22.094	Passed	

Figure 5.7: Testing Result for Reservation Management

5.2.3 System Testing Overall Result

The system testing is categorized to seven functions to test. Testing action is design to evaluate the criteria of the 24SDSMS. Each of them exercises a particular operating condition of the user's environment or features of the system, and will result in a pass or fail. The details of the Oracle Application Testing Suits action and result on 24SDSMS are shown in Appendix C. The sign off and acknowledgment of the software tester is shown in Appendix D. The summary of the result of Oracle Application Testing Suits on 24SDSMS-member is shown in Figure 5.8 and test result for administrator is show in Figure 5.9

Script Name: 24SDSMS System Testing

Report Generated By: oracle.oats.scripting.modules.functionalTest.api

Script Name: 24SDSMS System Testing

Workspace:

Date Time: 5/18/2012 18:37:29 PM

Iterations: 1

Total Steps: 32

Total User-Defined Tests: 6 **Passed:** 6 **Failed:** 0 **Warning:** 0

Total Script Actions: 73 **Passed:** 73 **Failed:** 0 **Warning:** 0

Total Passes: 79 (100.00%)

Total Failures: 0 (0.00%)

Total Warnings: 0 (0.00%)

Overall Result: Passed

Figure 5.8: Summary of the result of Oracle Application Testing Suits

Script Name: 24SDSMS System Testing-Admin

Report Generated By: oracle.oats.scripting.modules.functionalTest.api
Script Name: 24SDSMS System Testing-Admin
Workspace:
Date Time: 5/19/2012 00:40:48 AM

Iterations: 1
Total Steps: 14
Total User-Defined Tests: 21 **Passed:** 21 **Failed:** 0 **Warning:** 0
Total Script Actions: 63 **Passed:** 63 **Failed:** 0 **Warning:** 0

Total Passes: 84 (100.00%)
Total Failures: 0 (0.00%)
Total Warnings: 0 (0.00%)
Overall Result: Passed

Figure 5.9: Summary of the result of Oracle Application Testing Suits- Admin

5.3 Discussion and Analysis on the Outcomes

The objectives of 24SDSMS as following are successfully achieved:

- i) First objective is to develop an Society Management System. It was already achieved by developing the system completely with twenty modules as shown in Chapter 4 with print screen of each part of the system modules and coding to implement it.
- ii) Second objective is to manage reservation service. It was already achieved by developing this system with manage resources management and reservation management function that able to approve or reject the reservation made from the member online. And all the records of reservation from the member are kept in the database.
- iii) Third objective is to develop a System that matches all the requirement of president of 24 Season Drums Society.

The outcomes of the user acceptance testing and system testing show positive result to which all test actions passed. It proven that the acceptance criteria of the system are fulfilled with the requirement that set early on.

5.4 Constraints

The constraints during the development of the 24SDSMS are as following:

- i) The time constraint because need to allocate time for projects and assignments for other subjects too.

The constraints of the 24SDSMS are as following:

- i) Internet connection is needed to access the system.
- ii) Service will not be available if the server host of the system down.
- iii) Google Chrome web browser is strongly recommend use to display the system.
- iv) Member cannot register by themselves except register by administrator.
- v) Size of the profile picture uploaded cannot exceed 1024.
- vi) Member can only reserve 2 resources on same resource per day.
- vii) No email notification when reservation successfully made.

CHAPTER 6

CONCLUSION

As a conclusion, 24 Season Drums Society Management System is proposed to develop for automate the managing process of society. This system help to solve the problem faced by 24 Season Society which all society management mostly being done by manually and make the cost to acquire the society is relatively expensive. A new model combined from V model and iteration and incremental development approach are used to implement into the system for develop a very good understanding project. The project are separate into 5 incremental set and iteration process are apply in each incremental set for developer learning advantage from build to build which can improve the weakness from previous stage and allow developer to do any changes and add new functional capabilities according to customer requirement once any mistakes occur. This system is separated to three different types of users such as open user, member and administrator and the system is strictly controlled by the administrator. This system has 20 modules and main functions are member profile management, resources management, performance management and gallery management. After successfully developing this system, this system succeeds to achieve the objectives which are developing the 24season drums society management system using iterative & incremental approach and automate the managing process of 24 season drums society.

REFERENCES

- Adobe. 2011. What is Dreamweaver(online).
<http://www.adobe.com/products/dreamweaver/whatisdreamweaver> (19th Oct 2011)
- Boehm, B. 1988. *A Spiral Model of Software Development and Enhancement*, IEEE computer
- Bias,R., & Mayhew, D. 1994. *Cost – justifying usability*. (Eds). San Francisco: Morgon Kaufmann Publishers.
- Freetutes. 2007. The spiral life cycle model(online).
<http://www.freetutes.com/systemanalysis/sa2-spiral-model.html> (20 August 2010)
- Franklin,B. (n.d.). BrainyQuote.com (online).
<http://www.brainyquote.com/quotes/quotes/b/benjaminfr104457.html> (18 December 2011)
- George,A.M .1956. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Classics in the History of Psychology* (online). <http://psychclassics.yorku.ca/Miller/> (12th Oct 2011)
- Hong Xun Jiang., 2006. introduction. *Research on IT outsourcing based on IT systems management*, 5(1),1, viewed 9 OCT 2011, ACM Digital Library
- IBM developer Works. 2011. What is iterative development (online).
<http://www.ibm/developerworks/rational/library/may05/bittner-spence/>. (27 November 2011)
- Jonathan Arkell.2010. PHP language (online). <http://www.c2.com/cgi-bin/wiki?PhpLanguage> (19 OCT 2011)

- Malaysia Dog Club.2011a.Main Page (online).
<http://www.malaysiadogclub.com/> (19th Oct 2011)
- Malaysia Dog Club.2011b. Help / Reference (online).
<http://www.malaysiadogclub.com/helps-a-tips> (19 Oct 2011)
- Malaysia Dog Club.2011c.Articles Store (online).
<http://www.malaysiadogclub.com/dog-news> (19 Oct 2011)
- Michael Jackson Fan Club. 2011a. main page (online).
http://www.mjfanclub.net/home/index.php?option=com_content&view=section&id=11&Itemid=63(19th Oct 2011)
- Michael Jackson Fan Club.2011b. latest new (online).
http://www.mjfanclub.net/home/index.php?option=com_content&view=category&id=85&Itemid=82 (19 Oct 2011)
- Microsoft. 2011. Microsoft Visual Studio (online).
<http://www.microsoft.com/visualstudio/en-us> (19th Oct 2011)
- Mills, H. D., O'Neill, D. *et al.* 1980. The management of software engineering.
IBM Sys.J., 24(2), 414-77.(ch 3)
- MySQL.com. 2011. About MySQL (online). <http://www.mysql.com/about/>
 (19th Oct 2011)
- Netbeans. 2000. NetBeans IDE 6.9.1 Release Information (online).
<http://netbeans.org/community/releases/69/>(12 Oct 2011)
- Nielsen,J.,&Mack,R.L 1994. *Usability inspection methods*. New York: John Wiley & Sons.
- Nielsen,J.1993. Usability engineering. New York: AP Professional.
- Norman,D.A. 1998.*The design of everyday things*. New York:doubleday.
- Online Interactive Modules for Teaching Computer Science. 1997. The SpiralModel(online).<http://courses.cs.vt.edu/csonline/SE/Lessons/Spiral/index.html>(20th August 2010)

- Oracle. 2011a. JavaServer Pages Technology (online).
<http://java.sun.com/products/jsp/faq.html> (19th Oct 2011)
- Oracle. 2011b. Oracle Database Architecture (online).
http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/intro.htm(19th Oct 2011)
- Php.net. 2011. What is PHP (online). <http://www.php.net/>(19th Oct 2011)
- Ryals, L.,2005, "Making Customer Relationship Management Work: The Measurement and Profitable Management of Customer Relationships", *Journal of Marketing*, (69:4), pp. 252-261
- Royce, W. W.1970. Managing the development of large software systems: concepts and techniques. Proc. *IEEE WESTCON*, Los Angeles, CA.(ch. 3)
- Shneiderman, B., & Plaisant, C.2005. *Designing the user interface – Strategies for effective human-computer interaction* (3rd ed.). Reading, MA:addison-Wesley
- Shneiderman, B.1998. *Designing the user interface: Strategies for effective human- computer interaction* (3rd ed). Reading, MA: Addison Wesley.-
- Software Testing Process.2011a. incremental and Iterative development (online).<http://www.softwaretestingprocess.com/testmethod/incremental.html>(27 November 2011)
- Sommerville,A .2001a.Ain Sommerville Spftware Engineering (6th ed.). Software Processes. University of Strathclyde: Addison-Wesley
- Tonyfaull. 2011. Visualization (computer graphics)(online).
<http://en.wikipedia.org/wiki/Visualization> (1 December 2011)
- Vredenburg,K.,Mao,J.,Smith,P.W.,&Carey,T.2002. *A survey of user-centered design practice*. Paper read at Computer Human Interaction (CHI)

Webopedia,2011. .NET Framework

(online). http://www.webopedia.com/TERM/D/dot_NET_Framework.html (19 Oct 2011)

Wikipedia. 2011. ASP.net (online). <http://en.wikipedia.org/wiki/ASP.NET> (18 Oct 2011)

Wikipedia.2011a. Programming Language (online).
http://en.wikipedia.org/wiki/Programming_language (18 Oct 2011)

Wikipedia. 2011b. .NET Framework (online).
http://en.wikipedia.org/wiki/Integrated_development_environment (19 Oct 2011)

Wikipedia. 2011c. Adobe Dreamweaver (online).
http://en.wikipedia.org/wiki/Adobe_Dreamweaver (10 Oct 2011)

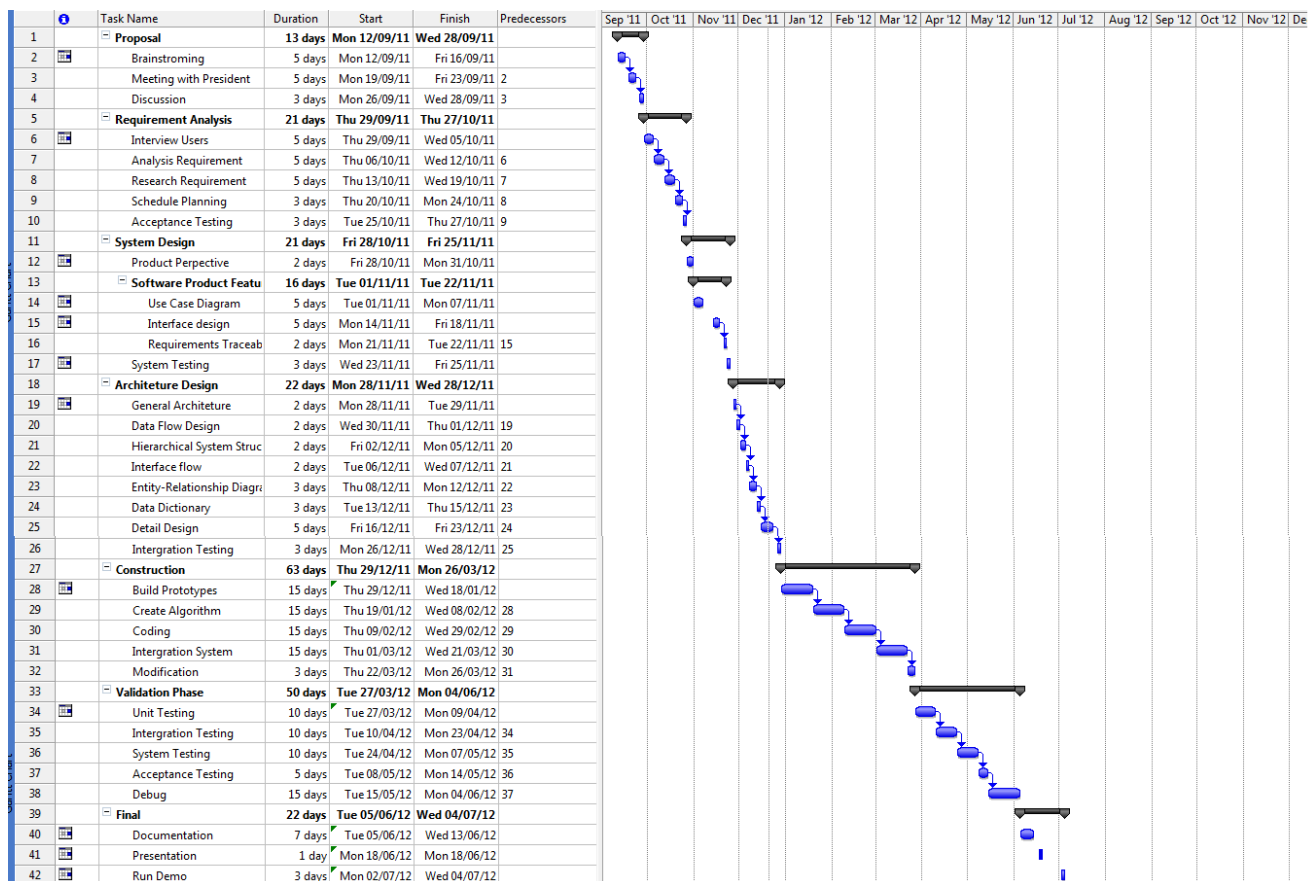
Wikipedia.2011d. Microsoft Visual Studio (online).
http://en.wikipedia.org/wiki/Microsoft_Visual_Studio(19 Oct 2011)

Wikipedia. 2011e. Database (online). <http://en.wikipedia.org/wiki/Database> (18 Oct 2011)

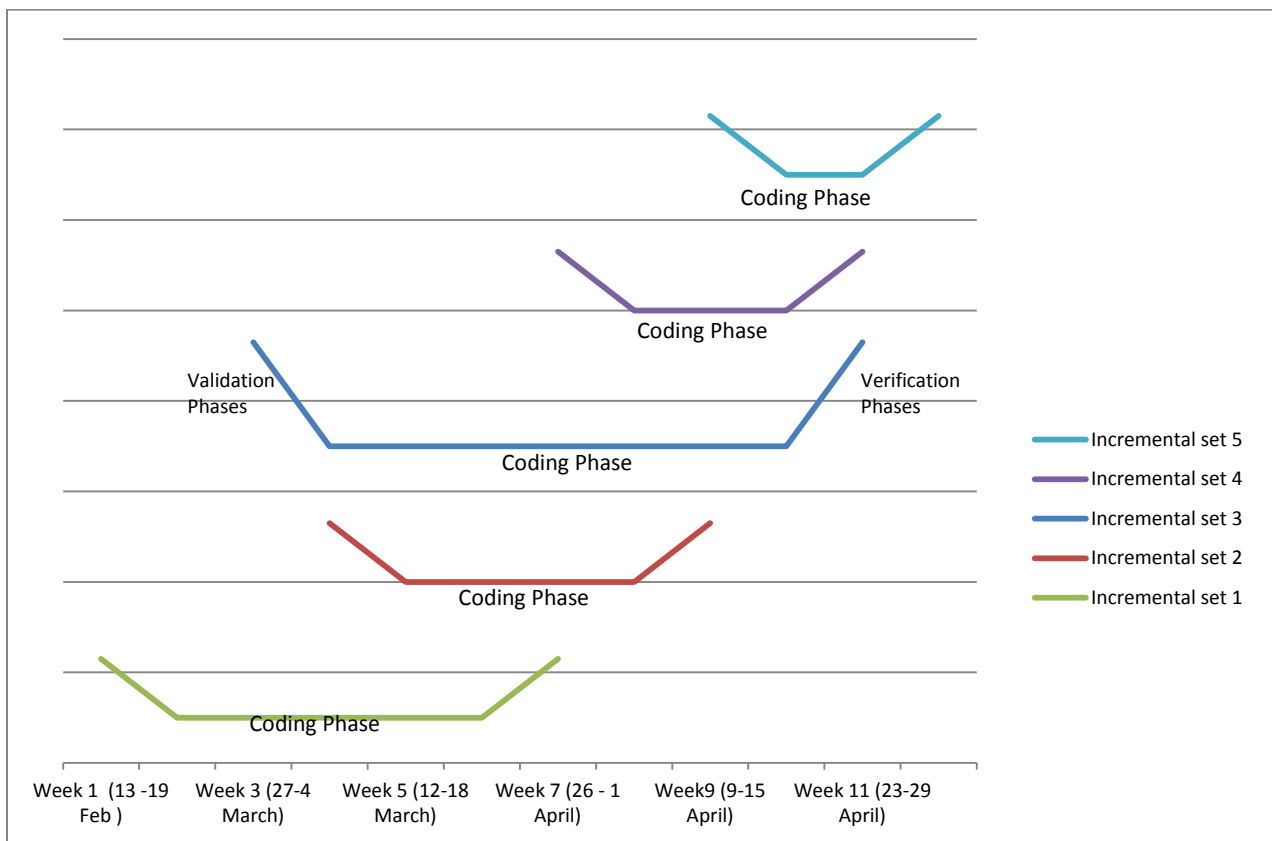
Wikipedia. 2011f. Microsoft SQL server (online).
http://en.wikipedia.org/wiki/Microsoft_SQL_Server (19th Oct 2011)

Appendix A

Gantt chart



A1- Gantt chart for whole project



A2- Gantt chart for each incremental

Appendix B

Implementation (coding)

```

<!--dbase.php-->
<?php

define("DATABASE_HOST","fskcs.ump.edu.my");
define("DATABASE_USER","cb09041");
define("DATABASE_PASSWORD","09041");

$conn = mysql_connect("fskcs.ump.edu.my","cb09041","09041");

if(!$conn){
    die("could not connect to database");
}

mysql_select_db("cb09041",$conn) or die ("could not open products database");

date_default_timezone_set('Asia/Kuala_Lumpur');

?>

```

B1: Database and Server Connection

```

<?php

//Start session
session_start();

//Include database connection details
require_once('config.php');

//Array to store validation errors
$msg_err = array();

//Validation error flag
$error_flag = false;

//Connect to mysql server
$link = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
if(!$link) {
    die('Failed to connect to server: ' . mysql_error());
}

//Select database
$db = mysql_select_db(DB_DATABASE);
if(!$db) {
    die("Unable to select database");
}

//Function to sanitize values received from the form. Prevents SQL injection
function clean($str) {
    $str = @trim($str);
    if(get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
    return mysql_real_escape_string($str);
}

//Sanitize the POST values
$username = clean($_POST['login']);
$password = clean($_POST['password']);
$nickname = clean($_POST['admin']);
//Input Validations
if ($nickname=='1'){
if($username == "") {
    $msg_err[] = 'Login ID missing';
    $error_flag = true;
}
if($password == "") {
    $msg_err[] = 'Password missing';
    $error_flag = true;
}

//If there are input validations, redirect back to the login form
if($error_flag) {
    $_SESSION['ERRMSG_ARR'] = $msg_err;
    session_write_close();
    header("location: login-failed.php");
    exit();
}

//Create query

$query="SELECT * FROM member_profile WHERE login='".$username"' AND passwd='".md5($_POST['password'])."' AND nickname='".$nickname'";
$result=mysql_query($query);
//Check whether the query was successful or not
if($result) {
    if(mysql_num_rows($result) == 1) {
        //Login Successful
        session_regenerate_id();
        $member = mysql_fetch_assoc($result);
        $_SESSION['SESS_MEMBER_ID_ADMIN'] = $member['member_id'];
        $_SESSION['SESS_FIRST_NAME_ADMIN'] = $member['firstname'];
        $_SESSION['SESS_LAST_NAME_ADMIN'] = $member['lastname'];
        session_write_close();
        header("location: ../admin/admin.php");
        exit();
    }
    }else {
        //Login failed
        header("location: login-failed.php");
        exit();
    }
}
} else {
    die("Query failed");
}
} else{
    if($username == "") {
        $msg_err[] = 'Login ID missing';
        $error_flag = true;
    }
}

```

```

    }
    if($password == "") {
        $errmsg_arr[] = 'Password missing';
        $errflag = true;
    }

    //If there are input validations, redirect back to the login form
    if($errflag) {
        $_SESSION['ERRMSG_ARR'] = $errmsg_arr;
        session_write_close();
        header("location: login-failed.php");
        exit();
    }

    //Create query

    $qry="SELECT * FROM member_profile WHERE login='$login' AND passwd='".md5($_POST['password'])."' AND nickname='$admin'";
    $result=mysql_query($qry);
    //Check whether the query was successful or not
    if($result) {
        if(mysql_num_rows($result) == 1) {
            //Login Successful
            session_regenerate_id();
            $member = mysql_fetch_assoc($result);
            $_SESSION['SESS_MEMBER_ID'] = $member['member_id'];
            $_SESSION['SESS_FIRST_NAME'] = $member['firstname'];
            $_SESSION['SESS_LAST_NAME'] = $member['lastname'];
            session_write_close();
            header("location: ../login.php");
            exit();
        } else {
            //Login failed
            header("location:login-failed.php");
            exit();
        }
    } else {
        die("Query failed");
    }
}
?>

```

B2: Login Function Coding

```

<!--kemaskini.php-->
<!--to update data of ubah.php into database.-->

<?php
include ("member_login/dbase.php");

extract($_POST);

$query = "UPDATE member_profile SET firstname='$fname', lastname='$lname', email='$email',
phone_no='$phone_no', street='$street',City='$city', Postcode='$postcode', login='$login',passwd='".md5($_POST['password'])."', P_picture="" . mysql_real_escape_string($newname) . ""
WHERE member_id='$id'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='Profile.php?id=$id'</script>";
}
?>

```

B3: 24SDSMS Edit Member profile – member coding

```

<?php

include("member_login/dbase.php");
extract($_POST);
//define a maxim size for the uploaded images in Kb
define ("MAX_SIZE", "1000");

//This function reads the extension of the file. It is used to determine if the file is an image by checking the extension.
function getExtension($str) {
    $i = strpos($str, ".");
    if ($i) { return ""; }
    $l = strlen($str) - $i;
    $ext = substr($str,$i+1,$l);
    return $ext;
}

//This variable is used as a flag. The value is initialized with 0 (meaning no error found)
//and it will be changed to 1 if an error occurs.
//If the error occurs the file will not be uploaded.
$errors=0;
//checks if the form has been submitted
if(isset($_POST['Submit']))
{
    //reads the name of the file the user submitted for uploading
    $image=$_FILES['image']['name'];
    //if it is not empty
    if ($image)
    {
        //get the original name of the file from the clients machine
        $filename = stripslashes($_FILES['image']['name']);
        //get the extension of the file in a lower case format
        $extension = getExtension($filename);
        $extension = strtolower($extension);

        //if it is not a known extension, we will suppose it is an error and will not upload the file,
        //otherwise we will do more tests
        if (($extension != "jpg" && ($extension != "jpeg") && ($extension != "png") && ($extension != "gif"))
        {
            //print error message
            echo '<h1>Unknown extension!</h1>';
            $errors=1;
        }
        else
        {
            //get the size of the image in bytes

```

```

//$_FILES['image']['tmp_name'] is the temporary filename of the file
//in which the uploaded file was stored on the server
$size=filesize($_FILES['image']['tmp_name']);

//compare the size with the maxim size we defined and print error if bigger
if ($size > MAX_SIZE*1024)
{
    echo '<h1>You have exceeded the size limit!</h1>';
    $errors=1;
}

//we will give an unique name, for example the time in unix time format
$image_name=time().'.'.$extension;
//the new name will be containing the full path where will be stored (images folder)

$image_name=time().'.'.$extension;
//the new name will be containing the full path where will be stored (images folder)
$newname="admin/members/".$image_name;

//we verify if the image has been uploaded, and print error instead
$scopied = copy($_FILES['image']['tmp_name'], $newname);
if (!$scopied)
{
    echo '<h1>Copy unsuccessful!</h1>';
    $errors=1;
}
}

//If no errors registered, print the success message
if(isset($_POST['Submit']) && !$errors)
{
    echo "<h1>File Uploaded Successfully! Try again!</h1>";
}

$newname2="members/$image_name";
$query = "UPDATE member_profile SET
    P_picture='".$mysql_real_escape_string($newname2)."' WHERE member_id='".$Sid"'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php");
if($result){
    echo "<script type='text/javascript'> window.location='Profile.php?id=$id'</script>";
}
?>

```

B4: 24SDSMS upload member picture profile coding

```

<h2>All Event/Training</h2><hr>
<?php
    $query = " SELECT * FROM event_training";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["id"];
        $name = $row["name"];
        $date = $row["date"];
        $time = $row["time"];

        ?>

        <li style="width:350px" ><a href="attendance2.php?id=<?php echo ($id); ?>" title="<?php echo $name; ?>"><span style="color:black;font-size:12px"><?php echo $name; ?></a><h5
        style="color:black;">date:<?php echo ($date); ?> | time:<?php echo ($time); ?></h5></li><br>

        <?php
        }
        ?>
    </ol>
<br>
<h2> Event/Training you have attended</h2><hr>
<?php

    $id=$_SESSION['SESS_MEMBER_ID'];
    $query=" SELECT * FROM member_profile WHERE member_id ='$id'";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
    $row = mysql_fetch_array($result, MYSQL_BOTH);

    $login =$row['login'];

    @mysql_free_result($result);
    ?>
<?php

    $query = " SELECT name,id,date,time FROM event_training, attendance_list WHERE event_id=id AND matrices_id='$login'";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["id"];
        $name = $row["name"];
        $date = $row["date"];
        $time = $row["time"];

        ?>

        <li style="width:350px" ><a href="attendance2.php?id=<?php echo ($id); ?>" title="<?php echo $name; ?>"><span style="color:black;">date:<?php echo ($date); ?> |
        time:<?php echo ($time); ?></h5></li><br>

        <?php
        }
        ?>
    </ol>

```

B5: 24SDSMS Attendance List – Member coding

```

<?php
include("dbase_member.php");
session_start();

//Array to store validation errors
$errmsg_arr = array();

//Validation error flag
$errflag = false;

function clean($str) {
    $str = @trim($str);
    if(get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
    return mysql_real_escape_string($str);
}

extract( $_POST);
$sm_id=$_SESSION['SESS_MEMBER_ID'];

$date =date("y-m-d",time());
$time =date("H:i:s",time());

$query = " SELECT * FROM resourcesList WHERE resources_list='$resources'";

$result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
$rows = mysql_fetch_array($result, MYSQL_BOTH);

    $sid=$row["l_id"];
        $resources_list= $row["resources_list"];
        $l_quantity= $row["l_quantity"];

        @mysql_free_result($result);

$quantity_total = $l_quantity - $quantity;

$diff = abs(strtotime($date4) - strtotime($date3));

$years = floor($diff / (365*60*60*24));
$months = floor(($diff - $years * 365*60*60*24) / (30*60*60*24));
$days = floor(($diff - $years * 365*60*60*24 - $months*30*60*60*24)/ (60*60*24));

if($quantity == "") {
    $errmsg_arr[] = 'quantity missing';
    $errflag = true;
}

if($quantity > '2') {
    $errmsg_arr[] = 'Each member only allowed to reserve 2 resources';
    $errflag = true;
}

if($date3 == '0000-00-00') {
    $errmsg_arr[] = 'Please insert your date';
    $errflag = true;
}

if($quantity > $l_quantity) {
    $errmsg_arr[] = 'Exceed quantity resources stored ';
    $errflag = true;
}

if($days > '2') {
    $errmsg_arr[] = 'Reservation period cannot more than 3 days';
    $errflag = true;
}

$query = "SELECT * FROM member_profile WHERE member_id='$sm_id'AND level='banned'";
$result = mysql_query($query);

if($result) {
    if(mysql_num_rows($result) > 0) {
        $errmsg_arr[] = 'Error: Your account have been banned, Please return Your Collected Resources As Soon As Possible';
        $errflag = true;
    }
    @mysql_free_result($result);
}

$query = "SELECT * FROM resources WHERE m_id='$sm_id'AND resources='$resources' AND reserved_date='$date'";
$result = mysql_query($query);

if($result) {
    if(mysql_num_rows($result) > 0) {
        $errmsg_arr[] = 'Error:Sorry, you can only reserve for one time on each resource per day';
        $errflag = true;
    }
    @mysql_free_result($result);
}

if($errflag) {
    $_SESSION['ERRMSG_ARR'] = $errmsg_arr;
    session_write_close();
    header("location: ResourcesList2.php?id=$sid ");
    exit();
}

$query = "INSERT INTO resources VALUES('$sm_id','$sid','$name','$matrics_id','$date3','$date4','$resources','$quantity','$date','$time','$status','','$') ";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");

$query = "UPDATE resourceslist SET L_quantity=$quantity_total WHERE l_id=$sid";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php11");

if($result){
    echo "<script type='text/javascript'> window.location='ResourcesList2-confirm.php'</script>";
}
?>

```

B6: 24SDSMS Resources Reservation Coding – Member

```

<?php
include("dbase_member.php");

extract( $_POST);
$tarikh =date("d-m-y",time());
$smasa = date("H:i:s",time());

$query = "INSERT INTO new VALUES(',$id','$stitle','$,$tarikh','$smasa)";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");
if($result){
    echo "<script type='text/javascript'>window.location=P_N.php' </script>";
}
?>

```

B7: 24SDSMS Post new – Member : coding

```

<?php
include("dbase_member.php");

extract( $_POST);
$tarikh =date("d-m-y",time());
$smasa = date("H:i:s",time());

$query = "INSERT INTO comment VALUES(',$New_id','$id','$comment','$tarikh','$smasa)";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");
if($result){
    echo "<script type='text/javascript'>window.location=P_N2.php?id=$New_id' </script>";
}
?>

```

B8: 24SDSMS reply comments – Member : coding

```

<h2>Performance List</h2><hr/>

<div style="height:600px;width:750px;overflow:auto;margin:0px auto;">
<?php

    $query = " SELECT * FROM performance ORDER BY p_id DESC";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["p_id"];
        $performance = $row["performance"];
        $description = $row["description"];
        $date = $row["date"];
        $time = $row["time"];

    }

<ol>
    <li style="width:350px"><a style="text-decoration:underline;color:#C30" href="P_T2.php?id=<?php echo ($id); ?>" title="<?php echo $performance;?>"><?php echo $performance;?></a>
    <br/><br/><article>Description:<?php echo $description ?></article><br/>
    <h5 style="color:black;">date:<?php echo ($date); ?> | time:<?php echo ($time); ?><br/></li>

    <?php
    }
?>

</ol>

</div>

```

B9: 24SDSMS Performance Timetable– Member: coding

```

<h2>performers List</h2><br/>
<center><hr /><a href="#" javascript:javascript:history.go(-1)">Click here to go back </a><hr/><br/>
<table style="color:black" border="2" width="500px">

<?php

    $performance_id= $_GET['id'];

    $query= "SELECT * FROM performance WHERE p_id = $performance_id";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
    $row = mysql_fetch_array($result, MYSQL_BOTH);

    $performance = $row['performance'];

    @mysql_free_result($result);
    ?>

    <h2> performance</h2><h2><?php echo $performance?></h2><br/>

<?php

    $query = " SELECT member_id,id,firstname,lastname,matrics_id FROM performer, member_profile WHERE login=matrics_id AND performance_id=$performance_id";

    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id=$row["id"];
        $firstname= $row["firstname"];
        $lastname= $row["lastname"];
        $m_id= $row["member_id"];
        $matrics_id = $row["matrics_id"];
    }
}
?>

```



```

        ?>
    <tr>
    <td><a href="Profile_other.php?id=<?php echo $m_id ?>"><?php echo ($firstname); ?> <?php echo ($lastname); ?></a></td><td><?php echo ($matrics_id); ?></td>
    </tr>
    <tr>
    <td>
    <?php
    }
    ?>
</table>

```

B10: 24SDSMS Performers Name List– Member: coding

```

<div style="height:200px;width:750px;overflow:auto;margin:0px auto;">
    <h2 style="margin-top:50px"> Your Today's Performance</h2><hr/>
    <?php

    $id=$_SESSION['SESS_MEMBER_ID'];
    $query="SELECT * FROM member_profile WHERE member_id ='$id";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
    $row = mysql_fetch_array($result, MYSQL_BOTH);

    $login =$row['login'];

    @mysql_free_result($result);
    ?>
<?php

    $query = " SELECT performance.p_id,description,date,time FROM performance, performer WHERE p_id=performance_id AND matrics_id='$login' AND `date` = CURDATE() ORDER BY
    p_id DESC";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["p_id"];
        $performance = $row["performance"];
        $description = $row["description"];
        $date = $row["date"];
        $time =$row["time"];

    ?>

<ol>
    <li style="width:350px"><a style="text-decoration:underline;color:#C30" href="P_T2.php?id=<?php echo ($id); ?>" title="<?php echo $performance;?>"><?php echo $performance;?></a>
    <br/><br/><article>Description:<?php echo $description ?></article><br/>
    <h5 style="color:black;">date:<?php echo ($date); ?> | time:<?php echo ($time); ?> <br/></li>

    <?php
    }
    ?>
</div>
</ol>

<div style="height:200px;width:750px;overflow:auto;margin:0px auto;">
    <h2 style="margin-top:20px"> Your Upcoming Performance</h2><hr/>
<?php

    $id=$_SESSION['SESS_MEMBER_ID'];
    $query="SELECT * FROM member_profile WHERE member_id ='$id";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
    $row = mysql_fetch_array($result, MYSQL_BOTH);

    $login =$row['login'];

    @mysql_free_result($result);
    ?>
<?php

    $query = " SELECT performance.p_id,description,date,time FROM performance, performer WHERE p_id=performance_id AND matrics_id='$login' AND `date` > CURDATE() ORDER BY
    p_id DESC";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["p_id"];
        $performance = $row["performance"];
        $description = $row["description"];
        $date = $row["date"];
        $time =$row["time"];

    ?>

<ol>
    <li style="width:350px"><a style="text-decoration:underline;color:#C30" href="P_T2.php?id=<?php echo ($id); ?>" title="<?php echo $performance;?>"><?php echo $performance;?></a>
    <br/><br/><article>Description:<?php echo $description ?></article><br/>
    <h5 style="color:black;">date:<?php echo ($date); ?> | time:<?php echo ($time); ?> <br/></li>

    <?php
    }
    ?>
</ol>
</script></div>

<div style="height:300px;width:750px;overflow:auto;margin:0px auto;">
    <h2> All of Your Performances</h2><hr/>
    <?php

    $id=$_SESSION['SESS_MEMBER_ID'];
    $query="SELECT * FROM member_profile WHERE member_id ='$id";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");

```

```

$row = mysql_fetch_array($result, MYSQL_BOTH);

    $login = $row['login'];

    @mysql_free_result($result);
    ?>
</php>

$query = " SELECT performance.p_id,description,date,time FROM performance, performer WHERE p_id=performance_id AND matrices_id=$login' ORDER BY p_id DESC";
$result= mysql_query($query,$conn);

while($row = mysql_fetch_array($result)){
    $id = $row["p_id"];
    $performance = $row["performance"];
    $description = $row["description"];
    $date = $row["date"];
    $time = $row["time"];

    ?>

<ol>
    <li style="width:350px"><a style="text-decoration:underline;color:#C30" href="P_T2.php?id=<?php echo ($id); ?>" title="<?php echo $performance;?>"><?php echo $performance;?></a>
<br/><br/><article>Description:<?php echo $description ?></article><br/>
<h5 style="color:black;">date:<?php echo ($date); ?> | time:<?php echo ($time); ?> <br/></li>

    <?php
    }
    ?>

</ol>

```

B11: 24SDSMS Individual Timetable– Member: coding

```

<h2>performers List</h2><br/>
<center><hr /><A HREF="#" javascript:javascript:history.go(-1)">Click here to go back </A><hr/><br/><br/>
<table style="color:black" border="2" width="500px">

<?php

    $performance_id= $_GET['id'];

    $query= "SELECT * FROM performance WHERE p_id = $performance_id";
    $result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
    $row = mysql_fetch_array($result, MYSQL_BOTH);

    $performance = $row['performance'];

    @mysql_free_result($result);
    ?>

    <h2> performance</h2><h2><?php echo $performance;?></h2><br/>
</php>

    $query = " SELECT member_id,id,firstname,lastname,matrics_id FROM performer, member_profile WHERE login=matrics_id AND performance_id=$performance_id";

    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id=$row["id"];
        $firstname= $row["firstname"];
        $lastname= $row["lastname"];
        $m_id= $row["member_id"];
        $matrics_id = $row["matrics_id"];

        ?>

<tr>
<td><a href="Profile_other.php?id=<?php echo $m_id ?>"><?php echo ($firstname); ?> <?php echo ($lastname); ?></a></td><td><?php echo ($matrics_id); ?></td>
</tr>

    <?php
    }
    ?>

</table>

```

B12: 24SDSMS Performers Name List– Member: coding

```

<h2>Album List</h2>
<?php

    $query = " SELECT * FROM album";
    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row["album_id"];
        $name = $row["name"];
        $url = $row["album_url"];
        $date = $row["tarikh"];
        $time = $row["masa"];

        ?>

<ol>
    <li class="indent" ><a href="picture_gallery.php?id=<?php echo ($id); ?>"><img src = "admin/<?php echo $url; ?>" title="<?php echo $name;?>" width = "130" Height="100" /></a>
<br/><center><a href="picture_gallery.php?id=<?php echo ($id); ?>" title="<?php echo $name;?>"><?php echo $name;?></a></center></li>

    <?php
    }
    ?>

</ol>

```

B13: 24SDSMS Albums– Member: Coding

```

<h2>Society Pictures</h2><h3> Album : </h3><hr />
<A HREF="javascript:javascript:history.go(-1)">Click here to go back </A><hr/><br/>

<?php
    $A_id = $_GET['id'];
    $Query = "SELECT * FROM pictures WHERE AL_id=$A_id";
    $result= mysql_query($Query,$conn);

    while($row = mysql_fetch_array($result)){
        $id = $row['id'];
        $AL_id = $row['AL_id'];
        $name = $row['name'];
        $uri = $row['uri'];
        $date = $row['date'];
        $time = $row['time'];

    }

    <li class="indent" ><a href="admin/?php echo ($uri); ?>" title="<?php echo $name;?>"><img src = "admin/?php echo $uri; ?>" width = "130" Height="100" </a><br/><center><?php echo $name;?></center></li>
<?php
}
?>

```

B14: 24SDSMS Pictures – Member: Coding

```

<?php
include("dbase_member.php");
extract($_POST);
$date =date("d-m-y",time());
$time = date("H:i:s",time());

$Query = "INSERT INTO complaint VALUES(',$category','$complaint','$date','$time) ";

$result=mysql_query($Query,$conn) or die("could not execute query in isikan.php");

if($result){
    echo "<script type='text/javascript'> window.location='thank_you.php'</script>";
}
?>

```

B15: 24SDSMS complaint – Member: Coding

```

<?php

$Query= "SELECT content FROM help WHERE h_id =1";
$result = mysql_query($Query, $conn) or die("could not execute query in ubah.php");
$row = mysql_fetch_array($result, MYSQL_BOTH);

$content= $row["content"];

@mysql_free_result($result);
?>

<article><?php echo $content ?></article>

```

B16: 24SDSMS help and information – Member: Coding

```

<?php
    $id=$_SESSION['SESS_MEMBER_ID'];
    $Query= "SELECT * FROM resources WHERE m_id = $id ORDER BY r_id DESC ";

    $result= mysql_query($Query,$conn);

    while($row = mysql_fetch_array($result)){
        $r_id = $row['r_id'];
        $name = $row['name'];
        $matrics_id = $row['matrics_id'];
        $date_from = $row['date_from'];
        $date_to = $row['date_to'];
        $resources = $row['resources'];
        $quantity = $row['quantity'];
        $reserved_date = $row['reserved_date'];
        $reserved_time = $row['reserved_time'];
        $status= $row['status'];

    }

    <table style="color:black" border="0" width="500px">
    <tr>
    <td>Confirmation Code</td><td><?php echo ($r_id); ?></td></tr>
    <tr>
    <td>Reserved Period</td><td><?php echo ($date_from); ?> - <?php echo ($date_to); ?></td></tr>
    <tr>
    <td>Resources</td><td><?php echo ($resources); ?></td></tr>
    <tr>
    <td>Quantity</td><td><?php echo ($quantity); ?></td></tr>
    <tr>
    <td>Reserved Date</td><td><?php echo ($reserved_date); ?></td></tr>
    <tr>
    <td>Reserved Time</td><td><?php echo ($reserved_time); ?></td></tr>
    <tr>
    <td>Status</td><td><?php echo ($status); ?></td></tr>

```

```

</tr>
        </table><hr/>
        <?php
        }
        ?>

```

B17: 24SDSMS Reserved History– Member: Coding

```

<h2> Advertisement List</h2><br/>
<?php
$query = "SELECT * from advertisement";

$result= mysql_query($query,$conn);

while($row = mysql_fetch_array($result)){
    $a_id=$row["id"];
    $icon = $row["icon"];
    $a_title = $row["title"];
    $stext = $row["text"];

    ?>
    <h3><a href="advertisement2.php?id=<?php echo $a_id; ?>"><?php echo $a_title;?></a></h3><br/>
</center>
<?php
}
?>

```

B18: 24SDSMS Advertisement List – Member: Coding

```

<?php

    $id = $_GET['id'];

    $query = "SELECT * from advertisement WHERE id = '$id'";

    $result= mysql_query($query,$conn);

    while($row = mysql_fetch_array($result)){
        $a_id=$row["id"];
                $icon = $row["icon"];
                $a_title = $row["title"];
                $stext = $row["text"];
    }
    ?>

    <center> <br/>
    <h2> <?php echo $a_title;?></h2><br/>
    <?php echo ($stext); ?><br/></center>

```

B19: 24SDSMS Advertisement Content – Member: Coding

```

<h2>Search Results</h2><hr /><A HREF="javascript:javascript:history.go(-1)">Click here to go back </A></hr/><br/>
        <?php
        extract( $_POST);

        if ($category=="member_profile"){
            $query="SELECT * FROM member_profile WHERE  firstname LIKE '% ' . $name . "% ' OR lastname LIKE '% ' . $name . "% ' OR login LIKE '% ' . $name . "% '";
            $result= mysql_query($query,$conn);

            while($row = mysql_fetch_array($result)){

                $member_id = $row["member_id"];
                $sname = $row["firstname"];
                $lname = $row["lastname"];
                $email = $row["email"];
                $phone_no = $row["phone_no"];
                $street = $row["street"];
                $city = $row["City"];
                $postcode = $row["Postcode"];
                $login = $row["login"];
                $passwd = $row["Passwd"];
                $P_picture = $row["P_picture"];

            }

            ?>

            <table width="350" border="0" cellpadding="2" cellspacing="0">

                <tr><td rowspan="9"><a href=Profile_other.php?id=<?php echo $member_id; ?>>" width="300px"></a> </td>

                <br/>

                <td colspan="2"> <h2>Profile Information</h2><br/></td>
            <tr>

                <th><a>First Name</a> </th>
                <td><?php echo $sname;?></td>
            </tr>
            <tr>
                <th><a>Last Name</a> </th>
                <td><?php echo $lname;?></td>
            </tr>
            <tr>
                <th><a>email</a> </th>
                <td><?php echo $email;?></td>
            </tr>
            <tr>
                <th width="124"><a>Phone Number</a></th>
                <td width="168"><?php echo $phone_no;?></td>
            </tr>
            <tr>
                <th width="124"><a>Address:Street</a></th>

```

```

<td width="168"><?php echo"$street";?></td>
</tr>
<tr>
<th width="124"><a>City</a></th>
<td width="168"><?php echo"$city";?></td>
</tr>
<tr>
<th width="124"><a>Postcode</a></th>
<td width="168"><?php echo"$postcode";?></td>
</tr>
<tr>
<th width="124"><a>Matrics ID</a></th>
<td width="168"><?php echo"$login";?></td>
</tr>
</table>
<?php
}
?>
<?php
}
?>
<?php
if ($category=="pictures"){
    $query="SELECT * FROM pictures WHERE name LIKE '%'. $name . '%";
    $result= mysql_query($query,$conn);
    while($row = mysql_fetch_array($result)){
    $id = $row["id"];
        $AL_id = $row["AL_id"];
        $name = $row["name"];
        $url = $row["url"];
        $date = $row["date"];
        $time = $row["time"];
        ?>
        <span style="text-indent: 0px; width: 17%; float:left; padding-top: 30px;" ><a href=" admin/<?php echo ($url); ?>" title="<?php echo $name; ?>"><img src = "admin/<?php echo $url; ?>" width
= "130" Height="100" <a><br/><center><?php echo $name; ?></center></span>
<?php
}
?>
<?php
}
?>
<?php
if ($category=="resources"){
    $id=$_SESSION['SESS_MEMBER_ID'];
    $query="SELECT * FROM resources WHERE resources LIKE '%'. $name . '% 'OR status LIKE '%'. $name . '% AND m_id = $id";
    $result= mysql_query($query,$conn);
    while($row = mysql_fetch_array($result)){
        $r_id = $row["r_id"];
        $name = $row["name"];
        $matrics_id = $row["matrics_id"];
        $date_from = $row["date_from"];
        $date_to = $row["date_to"];
        $resources = $row["resources"];
        $quantity = $row["quantity"];
        $reserved_date = $row["reserved_date"];
        $reserved_time = $row["reserved_time"];
        $status=$row["status"];
        ?>
        <table style="color:black" border="0" width="500px">
<tr>
<td>Confirmation Code</td><td><?php echo ($r_id); ?></td></tr>
<tr>
<td>Reserved Period</td><td><?php echo ($date_from); ?> - <?php echo ($date_to); ?></td></tr>
<tr>
<td>Name</td><td><?php echo ($name); ?></td></tr>
<tr>
<td>Matric ID</td><td><?php echo ($matrics_id); ?></td></tr>
<tr>
<td>Resources</td><td><?php echo ($resources); ?></td></tr>
<tr>
<td>Quantity</td><td><?php echo ($quantity); ?></td></tr>
<tr>
<td>Reserved Date</td><td><?php echo ($reserved_date); ?></td></tr>
<tr>
<td>Reserved Time</td><td><?php echo ($reserved_time); ?></td></tr>
<tr>
<td>Status</td><td><?php echo ($status); ?></td></tr>
</tr>
</table><br/>
<?php
}
?>
<?php
}
?>
</div></div></div></div><br/></center>
</body>
</html>

```

B20: 24SDSMS Search Result – Member: Coding

```

<?php
include("dbase_member.php");
extract( $_POST);
//define a maxim size for the uploaded images in Kb
define ("MAX_SIZE","1000");

//This function reads the extension of the file. It is used to determine if the file is an image by checking the extension.
function getExtension($sstr) {
    $i = strpos($sstr, ".");
    if (!$i) { return ""; }
    $l = strlen($sstr) - $i;
    $sext = substr($sstr,$i+1,$l);
    return $sext;
}

//This variable is used as a flag. The value is initialized with 0 (meaning no error found)
//and it will be changed to 1 if an error occurs.
//If the error occurs the file will not be uploaded.
$errors=0;
//checks if the form has been submitted
if(isset($_POST['Submit']))
{
    //reads the name of the file the user submitted for uploading
    $image=$_FILES['image']['name'];
    //if it is not empty
    if ($image)
    {
        //get the original name of the file from the clients machine
        $filename = stripslashes($_FILES['image']['name']);
        //get the extension of the file in a lower case format
        $extension = getExtension($filename);
        $extension = strtolower($extension);
        //if it is not a known extension, we will suppose it is an error and will not upload the file,
        //otherwise we will do more tests
        if (($extension != "jpg") && ($extension != "jpeg") && ($extension != "png") && ($extension != "gif"))
        {
            //print error message
            echo '<h1>Unknown extension!</h1>';
            $errors=1;
        }
        else
        {
            //get the size of the image in bytes
            //$_FILES['image']['tmp_name'] is the temporary filename of the file
            //in which the uploaded file was stored on the server
            $size=filesize($_FILES['image']['tmp_name']);

            //compare the size with the maxim size we defined and print error if bigger
            if ($size > MAX_SIZE*1024)
            {
                echo '<h1>You have exceeded the size limit!</h1>';
                $errors=1;
            }

            //we will give an unique name, for example the time in unix time format
            $image_name=time().'.'.$extension;
            //the new name will be containing the full path where will be stored (images folder)

            $image_name=time().'.'.$extension;
            //the new name will be containing the full path where will be stored (images folder)
            $newname="./images/".$image_name;

            //we verify if the image has been uploaded, and print error instead
            $copied = copy($_FILES['image']['tmp_name'], $newname);
            if (!$copied)
            {
                echo '<h1>Copy unsuccessful!</h1>';
                $errors=1;
            }
        }
    }
}

//If no errors registered, print the success message
if(isset($_POST['Submit']) && !$errors)
{
    echo "<h1>File Uploaded Successfully! Try again!</h1>";
}

$newname2="./cb09041/psm/images/$image_name";

$query = "UPDATE main_page SET
        url='".$mysql_real_escape_string($newname2)."' WHERE id='1'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='main_page_admin.php'</script>";
}
?>

```

B21: 24SDSMS Banner Edit –Administrator: Coding

```

<?php
include("dbase_member.php");
extract( $_POST);

$query = "UPDATE main_page SET
        Title='$title', text='$text' WHERE id='2'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='main_page_admin.php'</script>";
}
?>

```

B22: 24SDSMS Sidebar Edit –Administrator: Coding

```

<?php
include("dbase_member.php");
extract( $_POST);

$query = "UPDATE main_page SET
        Title='$title', text='$text' WHERE id='3'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
        echo "<script type='text/javascript'> window.location='main_page_admin.php'</script>";
}
?>

```

B23: 24SDSMS Top Content Edit –Administrator: Coding

```

<?php
include ("dbase_member.php");

extract($_POST);
$query = "UPDATE menu SET action='$Profile' WHERE name='Profile'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Attendance' WHERE name='Attendance'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Resources' WHERE name='Resources'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$News' WHERE name='News'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Post' WHERE name='Post'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Performance' WHERE name='Performance'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Individual' WHERE name='Individual'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Picture' WHERE name='Picture'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Video' WHERE name='Video'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Complaint' WHERE name='Complaint'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Donation' WHERE name='Donation'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Help' WHERE name='Help'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$History' WHERE name='History'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

$query = "UPDATE menu SET action='$Advertisement' WHERE name='Advertisement'";
$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");

if($result){
        echo "<script type='text/javascript'> window.location='main_page_admin.php'</script>";
}
?>

```

B24: 24SDSMS Menu Edit –Administrator: Coding

```

<?php
include("dbase_member.php");
extract( $_POST);
//define a maxim size for the uploaded images in Kb
define ("MAX_SIZE","1000");

//This function reads the extension of the file. It is used to determine if the file is an image by checking the extension.
function getExtension($str) {
        $i = strpos($str, ".");
        if ($i) { return "";}
        $l = strlen($str) - $i;
        $ext = substr($str,$i+1,$l);
        return $ext;
}

//This variable is used as a flag. The value is initialized with 0 (meaning no error found)
//and it will be changed to 1 if an error occurs.
//If the error occurs the file will not be uploaded.
$errors=0;
//checks if the form has been submitted
if(isset($_POST['Submit']))
{
        //reads the name of the file the user submitted for uploading
        $image=$_FILES['image']['name'];
        //if it is not empty
        if ($image)
        {
                //get the original name of the file from the clients machine
                $filename = stripslashes($_FILES['image']['name']);
                //get the extension of the file in a lower case format
                $extension = getExtension($filename);
                $extension = strtolower($extension);
                //if it is not a known extension, we will suppose it is an error and will not upload the file,
                //otherwise we will do more tests
                if (($extension != "jpg" )&& ($extension != "jpeg" ) && ($extension != "png" ) && ($extension != "gif" )
                {
                        //print error message

```

```

        echo <h1>Unknown extension!</h1>;
        $errors=1;
    }
    else
    {
//get the size of the image in bytes
//$_FILES['image']['tmp_name'] is the temporary filename of the file
//in which the uploaded file was stored on the server
$$size=filesize($_FILES['image']['tmp_name']);

//compare the size with the maxim size we defined and print error if bigger
if ($size > MAX_SIZE*1024)
{
    echo <h1>You have exceeded the size limit!</h1>;
    $errors=1;
}

//we will give an unique name, for example the time in unix time format
$image_name=time().'.'.$extension;
//the new name will be containing the full path where will be stored (images folder)

$image_name=time().'.'.$extension;
//the new name will be containing the full path where will be stored (images folder)
$$new_name= "./add/".$image_name;

//we verify if the image has been uploaded, and print error instead
$scopied = copy($_FILES['image']['tmp_name'], $$new_name);
if (!$scopied)
{
    echo <h1>Copy unsuccessful!</h1>;
    $errors=1;
}
}
}

//If no errors registred, print the success message
if (isset($_POST['Submit']) && !$errors)
{
    echo <h1>File Uploaded Successfully! Try again!</h1>;
}

$query = "INSERT INTO advertisement VALUES('".$_mysql_real_escape_string($new_name)."','".$title','$text') ";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");

if($result){
    echo "<script type='text/javascript'> window.location='advertisement.php'</script>";
}
}
?>

```

B25: 24SDSMS Advertisement Manager –Administrator: Coding

```

<?php include("dbase_member.php");

$id= $_GET['id'];

$query= "SELECT resources FROM resources WHERE l_id = '$id'";

$result= mysql_query($query,$conn);

$result = mysql_query($query, $conn) or die("could not execute query in ubah.php");
$row = mysql_fetch_array($result, MYSQL_BOTH);

$resources = $row['resources'];

@mysql_free_result($result);

?>

<h2>Reservation List - <?php echo $resources ?></h2><br/>
<center><a href="javascript:javascript:history.go(-1)">Click here to go back </a></center>

<h3>Collected</h3><br />
<table style="color:black" border="1" width="1000">
<tr>
<td>Confirmation Code</td>
<td>Name</td>
<td>Reservation Period</td>
<td>Quantity</td>
<td>Reserve Date & Time</td>
<td>Collect Date & Time</td>
<td>Status</td>
<td>Reputation </td>
</tr>

<h3 style="color:red">Out of Return Date</h3><br />
<table style="color:black" border="1" width="1000">
<tr>
<td>Confirmation Code</td>
<td>Name</td>
<td>Reservation Period</td>
<td>Quantity</td>
<td>Reserve Date & Time</td>
<td>Collect Date & Time</td>
<td>Return Date & Collect Time</td>
<td>Status</td>
<td>Reputation </td>
</tr>

</php

```



```

        $today = "date('y-m-d')";

        $query= "SELECT member_id,level_r_id,name,matrix_id,date_from,date_to,resources,quantity,reserved_date,reserved_time,collect_date,collect_time,status,return_date,return_time
FROM member_profile,resources WHERE l_id = '$id' AND member_id=m_id AND status='collected' AND date_to < " . date("Y-m-d") . " " ORDER BY r_id desc";

        $result= mysql_query($query,$conn);

        while($row = mysql_fetch_array($result)){
            $r_id = $row['r_id'];
            $member_id = $row['member_id'];
                $name = $row['name'];
            $matrix_id = $row['matrix_id'];
            $date_from = $row['date_from'];
            $date_to = $row['date_to'];
            $resources = $row['resources'];
            $quantity = $row['quantity'];
                $reserved_date = $row['reserved_date'];
                $reserved_time = $row['reserved_time'];
                $collect_date = $row['collect_date'];
                $collect_time = $row['collect_time'];
                    $return_date = $row['return_date'];
                $return_time = $row['return_time'];
            $status = $row['status'];
            $level = $row['level'];

            ?>

<tr>
<td><?php echo ($r_id); ?></td>
><td><a style="color:black" href="member_profile.php?id=<?php echo $member_id; ?>"><?php echo $name; ?></td>
<td><?php echo ($date_from); ?> - <?php echo ($date_to); ?></td>

<td><?php echo ($quantity); ?></td>
<td><?php echo ($reserved_date); ?> | <?php echo ($reserved_time); ?></td>
<td><?php echo ($collect_date); ?> | <?php echo ($collect_time); ?></td>
<td><?php echo ($return_date); ?> | <?php echo ($return_time); ?></td>
<td><?php echo ($status); ?></td>

<?php
$color = "#000000";
if ($level == 'Good')
    $color = "green";
else if ($level == 'Banned')
    $color = "red";
?>
<td style="color:green"><a style="color:<?php echo $color; ?>" href="reputation.php?id=<?php echo $member_id; ?>"><?php echo $level; ?></td>;

</tr>

<?php
}
?>
</table><br/>

```

B26: 24SDSMS Reservation Manager –Administrator Coding

```

<?php
//Start session
session_start();

//Include database connection details
require_once('config_member.php');

//Array to store validation errors
$errmsg_arr = array();

//Validation error flag
$errorflag = false;

//Connect to mysql server
$link = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
if(!$link) {
    die("Failed to connect to server: " . mysql_error());
}

//Select database
$db = mysql_select_db(DB_DATABASE);
if(!$db) {
    die("Unable to select database");
}

//Function to sanitize values received from the form. Prevents SQL injection
function clean($str) {
    $str = @trim($str);
    if(get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
    return mysql_real_escape_string($str);
}

//Sanitize the POST values
$name = clean($_POST['name']);
$name = clean($_POST['name']);
$email = clean($_POST['email']);
$phone_no = clean($_POST['phone_no']);
$street = clean($_POST['street']);
$city = clean($_POST['city']);
$postcode = clean($_POST['postcode']);
$username = clean($_POST['login']);
$password = clean($_POST['password']);
$cpassword = clean($_POST['cpassword']);
$nickname = clean($_POST['nickname']);

//Input Validations
if($name == "") {
    $errmsg_arr[] = 'First name missing';
    $errorflag = true;
}
if($name == "") {

```

```

        Serrmsg_arr[] = 'Last name missing';
        Serrflag = true;
    }
    if($email == "") {
        Serrmsg_arr[] = 'Email missing';
        Serrflag = true;
    }
    if($phone_no == "") {
        Serrmsg_arr[] = 'phone number missing';
        Serrflag = true;
    }
    if($street == "") {
        Serrmsg_arr[] = 'city missing';
        Serrflag = true;
    }
    if($city == "") {
        Serrmsg_arr[] = 'Postcode missing';
        Serrflag = true;
    }
    if($postcode == "") {
        Serrmsg_arr[] = 'Postcode missing';
        Serrflag = true;
    }
    if($login == "") {
        Serrmsg_arr[] = 'Login ID missing';
        Serrflag = true;
    }
    if($password == "") {
        Serrmsg_arr[] = 'Password missing';
        Serrflag = true;
    }
    if($cpassword == "") {
        Serrmsg_arr[] = 'Confirm password missing';
        Serrflag = true;
    }
    if( strcmp($password, $cpassword) != 0 ) {
        Serrmsg_arr[] = 'Passwords do not match';
        Serrflag = true;
    }
}

//Check for duplicate login ID
if($login != "") {
    $qry = "SELECT * FROM member_profile WHERE login='$login' AND nickname='$nickname'";
    $result = mysql_query($qry);
    if($result) {
        if(mysql_num_rows($result) > 0) {
            Serrmsg_arr[] = 'Login ID already in use';
            Serrflag = true;
        }
        @mysql_free_result($result);
    }
    else {
        die("Query failed");
    }
}

//If there are input validations, redirect back to the registration form
if($serrflag) {
    $_SESSION['ERRMSG_ARR'] = $serrmsg_arr;
    session_write_close();
    header("location: register-form.php");
    exit();
}

//Create INSERT query
$qry = "INSERT INTO member_profile(firstname, lastname, email, phone_no, street, City, Postcode, login, passwd, P_picture, nickname,level)
VALUES('$fname','$lname','$email','$phone_no','$street','$city','$postcode','$login','$md5($_POST['password'])','$P_picture','$nickname','Good')";
$result = @mysql_query($qry);

//Check whether the query was successful or not
if($result) {
    header("location: register-form.php");
    exit();
}
else {
    die("Query failed");
}
}
?>

```

B27: 24SDSMS Member Profile Manager –Administrator: Coding

```

<?php
include ("dbase_member.php");
extract($_POST);
$query = "UPDATE member_profile SET firstname='$fname', lastname='$lname', email='$email', level='$level',phone_no='$phone_no', street='$street',City='$city',
Postcode='$postcode', login='$login',passwd='$_md5($_POST['password'])', P_picture='$_mysql_real_escape_string($newname)' WHERE member_id='$id'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='member_profile.php?id=$id'</script>";
}
}
?>

```

B28: 24SDSMS Member Profile Manager –Administrator: Coding

```

<?php
include("dbase_member.php");
extract($_POST);
$query = "INSERT INTO event_training VALUES('$sname','$sdate','$stime') ";
$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");
if($result){

```

```

        echo "<script type='text/javascript'> window.location='attendance_m.php'</script>";
    }
?>

```

B29: 24SDSMS Insert New Attendance –Administrator : Coding

```

<?php
include("dbase_member.php");
extract( $_POST);

$query = "INSERT INTO resourcesList VALUES(',$resources_list','$l_quantity' )";
$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");
if($result){
    echo "<script type='text/javascript'> window.location='ResourcesManagement.php'</script>";
}
?>

```

B30: 24SDSMS Add New Resources–Administrator: Coding

```

<?php
include("dbase_member.php");

$id = $_GET['id'];

$query = "DELETE FROM resourcesList WHERE l_id='$id' ";
$result = mysql_query($query, $conn);

if($result){
    echo "<script type='text/javascript'>window.location='ResourcesManagement.php'</script>";
}
?>

```

B31: 24SDSMS Delete Resources–Administrator: Coding

```

<?php
include ("dbase_member.php");
extract($_POST);
$query = "UPDATE resourcesList SET resources_list=',$resources_list', l_quantity='$l_quantity' WHERE l_id='$l_id'";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='ResourcesManagement.php'</script>";
}
?>

```

B32: 24SDSMS Update Resources–Administrator: Coding

```

<?php include ("dbase_member.php");

session_start();
//Include database connection details
require_once('config_member.php');
//Array to store validation errors
$error_msg = array();
//Validation error flag
$error_flag = false;

extract($_POST);
$date = date("y-m-d",time());
$time = date("H:i:s",time());

if($code != "") {
    $query = "SELECT * FROM resources WHERE r_id='$code'AND status='collected'";
    $result = mysql_query($query);

    if($result) {
        if(mysql_num_rows($result) > 0) {
            $error_msg_arr[] = 'Error: Resources have been collected';
            $error_flag = true;
        }
        @mysql_free_result($result);
    }
}

$query = "SELECT * FROM resources WHERE r_id='$code'AND status='returned'";
$result = mysql_query($query);

if($result) {
    if(mysql_num_rows($result) > 0) {
        $error_msg_arr[] = 'Error: Invalid Code';
        $error_flag = true;
    }
    @mysql_free_result($result);
}

$query = "SELECT * FROM resources WHERE r_id='$code'";
$result = mysql_query($query);

if($result) {
    if(mysql_num_rows($result) < 1) {
        $error_msg_arr[] = 'Error: Invalid Code';
        $error_flag = true;
    }
    @mysql_free_result($result);
}
}

```

```

                else {
                    die("Query failed");
                }
            }

            //If there are input validations, redirect back to the registration form
            if($errflag) {
                $_SESSION['ERRMSG_ARR'] = $errmsg_arr;
                session_write_close();
                header("location: resourcesCollect.php");
                exit();
            }

            $query = "UPDATE resources SET status='collected', collect_date='$date', collect_time='$time' WHERE r_id='$code'";
            $result = mysql_query($query,$conn) or die("could not execute query in isikan.php11");

            if($result){
                echo "<script type='text/javascript'> window.location='resourcesCollect2.php?id=$code'</script>";
            }
        }
    }
}
?>

```

B33: 24SDSMS Collect Resources–Administrator: Coding

```

<?php include ("dbase_member.php");

session_start();

//Include database connection details
require_once('config_member.php');

//Array to store validation errors
$errmsg_arr = array();

//Validation error flag
$errflag = false;

extract($_POST);
$date = date("y-m-d",time());
$time = date("H:i:s",time());
if($code != "") {

    $query = "SELECT * FROM resources WHERE r_id='$code' AND status='returned'";
    $result = mysql_query($query);

    if($result) {
        if(mysql_num_rows($result) > 0) {
            $errmsg_arr[] = 'Error: Resources have been returned';
            $errflag = true;
        }
        @mysql_free_result($result);
    }

    $query = "SELECT * FROM resources WHERE r_id='$code' AND status='pending'";
    $result = mysql_query($query);

    if($result) {
        if(mysql_num_rows($result) > 0) {
            $errmsg_arr[] = 'Error: Resources have not collect yet';
            $errflag = true;
        }
        @mysql_free_result($result);
    }

    $query = "SELECT * FROM resources WHERE r_id='$code'";
    $result = mysql_query($query);

    if($result) {
        if(mysql_num_rows($result) < 1) {
            $errmsg_arr[] = 'Error: Invalid Code';
            $errflag = true;
        }
        @mysql_free_result($result);
    }

    else {
        die("Query failed");
    }
}

//If there are input validations, redirect back to the registration form
if($errflag) {
    $_SESSION['ERRMSG_ARR'] = $errmsg_arr;
    session_write_close();
}

```

```

        header("location: resourcesReturn.php");
        exit();
    }

    echo "<script type='text/javascript'> window.location='resourcesReturn2.php?id=$code'</script>";
?>

```

B34: 24SDSMS Return Resources–Administrator: Coding

```

<?php
include("dbase_member.php");
extract( $_POST);

$stime = $hour.":". $min.":". $sec;

$query = "INSERT INTO performance VALUES(','$performance','$description','$date','$time') ";
$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");

if($result){
    echo "<script type='text/javascript'> window.location='P_T_A.php'</script>";
}
?>

```

B35: 24SDSMS add Performance–Administrator: coding

```

<?php include ("dbase_member.php");

    session_start();

    //Include database connection details
    require_once('config_member.php');

    //Array to store validation errors
    $errmsg_arr = array();

    //Validation error flag
    $errflag = false;

extract($_POST);
if($matrics_id != "") {

    $query = "SELECT * FROM member_profile WHERE login='$matrics_id'";
    $result = mysql_query($query);
    if($result) {
        if(mysql_num_rows($result) < 1) {
            $errmsg_arr[] = 'Error: member does not exist';
            $errflag = true;
        }
        @mysql_free_result($result);
    }

    $query = "SELECT * FROM performer WHERE matrics_id='$matrics_id' AND performance_id='$performance_id'";
    $result = mysql_query($query);
    if($result) {
        if(mysql_num_rows($result) > 0) {
            $errmsg_arr[] = 'Error: matrics has been key in';
            $errflag = true;
        }
        @mysql_free_result($result);
    }
    else {
        die("Query failed");
    }
}

//If there are input validations, redirect back to the registration form
if($errflag) {
    $_SESSION['ERRMSG_ARR'] = $errmsg_arr;
    session_write_close();
    header("location: P_T_A2.php?id=$performance_id");
    exit();
}

$query = "INSERT INTO performer VALUES('$matrics_id','$performance_id') ";

$result=mysql_query($query,$conn) or die("could not execute query in isikan.php");

if($result){

    echo "<script type='text/javascript'> window.location='P_T_A2.php?id=$performance_id'</script>";

}
?>

```

B36: 24SDSMS add Performers–Administrator: coding

```

<?php
include("dbase_member.php");

//dapatkan tarikh dan masa masuk
extract( $_POST);
$tarikh = date("d-m-y",time());
$masa = date("H:i:s",time());

```

```

//table : userlogin
$query = "INSERT INTO album VALUES(',$album',,$starih',,$smasa)";

$result=mysql_query($query,$conn) or die ("could not execute query in kemaskini.php ");
if($result){
    echo "<script type='text/javascript'> window.location='admin-upload.php'</script>";
}
?>

```

B37: 24SDSMS Create Album–Administrator: Coding

```

<?php
include("dbase_member.php");

//dapatkan tarikh dan masa masuk
extract( $_POST);
$date =date("d-m-y",time());
$time = date("H:i:s",time());

//define a maxim size for the uploaded images in Kb
define ("MAX_SIZE","100");

//This function reads the extension of the file. It is used to determine if the file is an image by checking the extension.
function getExtension($str) {
    $i = strpos($str, ".");
    if ($i) { return ""; }
    $l = strlen($str) - $i;
    $ext = substr($str,$i+1,$l);
    return $ext;
}

//This variable is used as a flag. The value is initialized with 0 (meaning no error found)
//and it will be changed to 1 if an error occurs.
//If the error occurs the file will not be uploaded.
$errors=0;
//checks if the form has been submitted
if(isset($_POST['submit']))
{
    //reads the name of the file the user submitted for uploading
    $image=$_FILES['image']['name'];
    //if it is not empty
    if ($image)
    {
        //get the original name of the file from the clients machine
        $filename = stripslashes($_FILES['image']['name']);
        //get the extension of the file in a lower case format
        $extension = getExtension($filename);
        $extension = strtolower($extension);
        //if it is not a known extension, we will suppose it is an error and will not upload the file,
        //otherwise we will do more tests
        if (($extension != "jpg") && ($extension != "jpeg") && ($extension != "png") && ($extension != "gif"))
        {
            //print error message
            echo '<h1>Unknown extension!</h1>';
            $errors=1;
        }
        else
        {
            //get the size of the image in bytes
            //$_FILES['image']['tmp_name'] is the temporary filename of the file
            //in which the uploaded file was stored on the server
            $size=filesize($_FILES['image']['tmp_name']);

            //compare the size with the maxim size we defined and print error if bigger
            if ($size > MAX_SIZE*1024)
            {
                echo '<h1>You have exceeded the size limit!</h1>';
                $errors=1;
            }

            //we will give an unique name, for example the time in unix time format
            $image_name=time().'.'.$extension;
            //the new name will be containing the full path where will be stored (images folder)

            $image_name=time().'.'.$extension;
            //the new name will be containing the full path where will be stored (images folder)
            $newname="pictures/".$image_name;

            $copied = copy($_FILES['image']['tmp_name'], $newname);
            if (!$copied)
            {
                echo '<h1>Copy unsuccessful!</h1>';
                $errors=1;
            }
        }
    }
}

if(isset($_POST['submit']) && !$errors)
{
    echo "<h1>File Uploaded Successfully! Try again!</h1>";
}

$query = "UPDATE album SET album_url=' . mysql_real_escape_string($newname) . ' WHERE album_id=',$album_id'";
$result=mysql_query($query,$conn) or die ("could not execute query in isikan.php");
?>
<?php
$query = "INSERT INTO pictures VALUES(',$album_id',,$name', . mysql_real_escape_string($newname) . ',,$date',,$time) ";

$result=mysql_query($query,$conn) or die ("could not execute query in isikan.php");

if($result){
    echo "<script type='text/javascript'> window.location='admin-upload2.php?id=$album_id'</script>";
}
?>

```

B38: 24SDSMS Add Pictures –Administrator: Coding

```

<SCRIPT LANGUAGE="javascript">
function enter(evt)
    {
        var charCode = (evt.which) ? evt.which : window.event.keyCode;

        if (charCode == 13)
        {
            document.theform.submit();
        }
    }
}
</SCRIPT>
<?php include("dbase_member.php");

$query = " SELECT * FROM complaint ORDER BY c_id DESC";
$result= mysql_query($query,$conn);

while($row = mysql_fetch_array($result)){
    $id = $row["c_id"];
    $category = $row["category"];
    $complaint = $row["complaint"];
    $date = $row["date"];
    $time = $row["time"];
}

Time/Date : <?php echo"$date / $time";?><br/>
category:<?php echo $category; ?><br/>
complaint:<?php echo $complaint;?><br/>
<a style="color:blue" href="complaint_I_A-delete.php?id=<?php echo ($id); ?>" title="delete">Delete</a>

<br/>

<?php
}
?>

```

B39: 24SDSMS Complaint Management –Administrator: Coding

Appendix C

Acceptance Testing and System Testing

No	User Acceptance Testing Action	Pass/Fail	Date	Initials
	Major Functionality			
1	Do all modules work based on the requirement?	PASS	17/5/12	LCS
2	Are the company background and information introduced?	PASS	17/5/12	LCS
3	Does the system provide member login?	PASS	17/5/12	LCS
4	Does the member allow customizing their profile information?	PASS	17/5/12	LCS
5	Does the system provide individual performance view for member?	PASS	17/5/12	LCS
6	Does the system provide resources reservation for member?	PASS	17/5/12	LCS
7	Does the system provide forum platform for user?	PASS	17/5/12	LCS
8	Does the system provide Pictures and video gallery?	PASS	17/5/12	LCS
	Interface Appearance			
1	Does the system design appropriate for society member and open user?	PASS	17/5/12	LCS
2	Are all fonts, colors, shading and background consistent with standards in every interface?	PASS	17/5/12	LCS
3	Are all images match with the theme?	PASS	17/5/12	LCS
4	Are all header, footer, right and left panel match and consistent with standards in every interface?	PASS	17/5/12	LCS
	Different Level of Access			
1	Does the system provide society member and administrator login?	PASS	17/5/12	LCS
2	Do the system modules is only available for registered member only?	PASS	17/5/12	LCS
3	Do the system modules for management is only available for admin only?	PASS	17/5/12	LCS
4	Is the user restricted to reserve resource which controlled by admin?	PASS	17/5/12	LCS
	Performance			
1	Is the system webpage load in short period?	PASS	17/5/12	CPK
2	Is the system react fast when operation done by	PASS	17/5/12	CPK

	user?			
	Capacity			
1	Is there a capacity for the system to store data?	PASS	17/5/12	LCS
2	Is there a capacity to upload profile pictures?	PASS	17/5/12	LCS
	Accuracy			
1	Do the window titles correctly identify each module?	PASS	17/5/12	LCS
2	Does each link navigate to the correct module?	PASS	17/5/12	LCS
3	Are field labels appropriate?	PASS	17/5/12	LCS
4	Do all dates display correctly when making reservation and post comments?	PASS	17/5/12	LCS
5	Can you print the order details correctly?	PASS	17/5/12	LCS
6	Does the search bring up the expected results correctly?	PASS	17/5/12	LCS
7	Do all information listed correctly?	PASS	17/5/12	LCS
8	Are the search result display are match to what user search?	PASS	17/5/12	LCS
	Reliability			
1	Does the system recover from failure in a short time?	PASS	17/5/12	CPK
2	Does the system pop up alert message box to alert user when errors occur?	PASS	17/5/12	CPK
	Security			
1	Does the system need to key in password to login?	PASS	17/5/12	YSQ
2	Does the system direct access denied once detect unauthorized access?	PASS	17/5/12	YSQ
	Usability			
1	Does the system found ease to use?	PASS	17/5/12	YSQ
2	Do all modules have brief description?	PASS	17/5/12	YSQ
3	Is there hint text available for all applicable items?	PASS	17/5/12	YSQ
4	Does the system guide user in a proper way?	PASS	17/5/12	YSQ
	Exception Handling			
1	Are error, warning, information and confirmation messages accurate and understandable?	PASS	17/5/12	YSQ
2	Does the system provide validation and error handling when user input errors occur?	PASS	17/5/12	YSQ

3	Does the system disable the reservation once the quantity becomes zero?	PASS	17/5/12	YSQ
4	Does the system disallow the member to make reservation once banned by administrator?	PASS	17/5/12	YSQ

Script Name: 24SDSMS System Testing

Report Generated By: oracle.oats.scripting.modules.functionalTest.api
 Script Name: 24SDSMS System Testing
 Workspace:
 Date Time: 5/18/2012 18:37:29 PM

Iterations: 1
 Total Steps: 32
 Total User-Defined Tests: 6 Passed: 6 Failed: 0 Warning: 0
 Total Script Actions: 73 Passed: 73 Failed: 0 Warning: 0

Total Passes: 79 (100.00%)
 Total Failures: 0 (0.00%)
 Total Warnings: 0 (0.00%)
 Overall Result: Passed

Script Summary

Section	Name	Duration (sec)	Result	Summary
Initialize	Initialize Total (sec)	0.831	Passed	
	Launch Browser	0.716	Passed	
Iteration 1	Iteration Total (sec)	217.409	Passed	
	[1] No Title	0.664	Passed	
	[2] Login Failed (/login-failed.php)	10.916	Passed	
	[3] 24 Season Drums-main page (/login.php)	3.447	Passed	
	[4] Logged Out (/logout.php)	6.910	Passed	
	[5] 24 Season Drums-main page (/admin.php)	4.019	Passed	
	[6] Logged Out (/logout.php)	7.689	Passed	
	[7] 24 Season Drums-picture (/search.php)	5.598	Passed	
	[8] 24 Season Drums-picture (/search.php)	9.497	Passed	
	[9] 24 Season Drums-picture (/search.php)	5.895	Passed	
	[10] 24 Season Drums-picture (/search.php)	7.796	Passed	
	[11] 24 Season Drums-picture (/search.php)	6.176	Passed	
	[12] 24 Season Drums-News & Comments (/P_N.php)	4.303	Passed	
	[13] 24 Season Drums-News & Comments (/P_N.php)	8.815	Passed	
	[14] 24 Season Drums-News & Comments (/P_N2.php)	4.795	Passed	
	[15] 24 Season Drums-Resources List (/ResourcesList.php)	26.004	Passed	
	[16] 24 Season Drums-Resources List (/ResourcesList2.php)	5.773	Passed	
	[17] 24 Season Drums-Resources List (/calendar_form.php)	4.229	Passed	
	[18] 24 Season Drums-Resources List (/calendar_form.php)	3.802	Passed	
	[19] 24 Season Drums-Resources List (/ResourcesList2.php)	13.868	Passed	
	[20] 24 Season Drums-Resources List (/calendar_form.php)	2.661	Passed	
	[21] 24 Season Drums-Resources List (/calendar_form.php)	7.605	Passed	
	[22] 24 Season Drums-Resources List (/ResourcesList2.php)	4.959	Passed	
	[23] 24 Season Drums-Resources List (/calendar_form.php)	1.930	Passed	
	[24] 24 Season Drums-Resources List (/calendar_form.php)	4.797	Passed	
	[25] 24 Season Drums-Resources List (/ResourcesList2.php)	13.111	Passed	
	[26] 24 Season Drums-Resources List (/calendar_form.php)	2.066	Passed	
	[27] 24 Season Drums-Resources List (/calendar_form.php)	4.637	Passed	
	[28] 24 Season Drums-Resources List (/ResourcesList2-confirm.php)	7.747	Passed	
	[29] 24 Season Drums-main page (/ResourcesHistory.php)	6.332	Passed	
	[30] 24 Season Drums-Resources List (/ResourcesList.php)	1.920	Passed	
	[31] 24 Season Drums-Resources List (/ResourcesList2.php)	7.104	Passed	
	[32] 24 Season Drums-Resources List (/ResourcesList2.php)	1.605	Passed	
Finish	Finish Total (sec)	0.147	Passed	
	Script Total (sec)	218.602	Passed	

Test Results Summary

Total	Result	0%	100%
0	Failed	<div style="width: 0%; height: 10px; background-color: red;"></div>	
0	Warning	<div style="width: 0%; height: 10px; background-color: orange;"></div>	
79	Passed	<div style="width: 100%; height: 10px; background-color: green;"></div>	

Script Name: 24SDSMS System Testing-Admin

Report Generated By: oracle.nvts.scripting.modules.functionalTest.api
 Script Name: 24SDSMS System Testing-Admin
 Workspace:
 Date Time: 5/19/2012 00:40:48 AM

Iterations: 1
 Total Steps: 14
 Total User-Defined Tests: 21 Passed: 21 Failed: 0 Warning: 0
 Total Script Actions: 63 Passed: 63 Failed: 0 Warning: 0

Total Passes: 84 (100.00%)
 Total Failures: 0 (0.00%)
 Total Warnings: 0 (0.00%)
 Overall Result: **Passed**

Script Summary

Section	Name	Duration (sec)	Result	Summary
[-] Initialize	Initialize Total (sec)	0.801	Passed	
	Launch Browser	0.638	Passed	
[-] Iteration1	Iteration Total (sec)	116.757	Passed	
	[1] No Title	0.621	Passed	
	[2] 24 Season Drums-main page (/register-form.php)	2.685	Passed	
	[3] 24 Season Drums-main page (/register-form.php)	22.731	Passed	
	[4] 24 Season Drums-main page (/register-form.php)	14.018	Passed	
	[5] 24 Season Drums-Resources Management (/ResourcesManagement.php)	19.767	Passed	
	[6] 24 Season Drums-Resources Management (/ResourcesCollect.php)	2.658	Passed	
	[7] 24 Season Drums-Resources Management (/ResourcesCollect.php)	24.606	Passed	
	[8] 24 Season Drums-Resources Management (/ResourcesReturn.php)	4.695	Passed	
	[9] 24 Season Drums Resources Management (/ResourcesReturn.php)	2.937	Passed	
	[10] 24 Season Drums-Reservation Management (/ResourcesManagement3.php)	3.949	Passed	
	[11] 24 Season Drums-Reservation Management (/ResourcesManagement4.php)	8.068	Passed	
	[12] 24 Season Drums-Reservation Management (/ResourcesManagement3.php)	2.367	Passed	
	[13] 24 Season Drums-Reservation Management (/ResourcesManagement5.php)	3.154	Passed	
	[14] 24 Season Drums Reservation Management (/ResourcesManagement3.php)	0.577	Passed	
[-] Finish	Finish Total (sec)	0.115	Passed	
	Script Total (sec)	117.918	Passed	

Test Results Summary

Total	Result	0%	100%
0	Failed		
0	Warning		
84	Passed		

APPENDIX D
SIGN OFF AND ACKNOWLEDGEMENT