**UMP** Confession Application

LIM KAH CHUN

# BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING) WITH HONOURS

# UNIVERSITI MALAYSIA PAHANG

# UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT		
Author's Full Name : LIM KAH CHUN		
Date of Birth		
Title : UMP Confession Application		
Academic Session : SEMESTER I ACADEMIC SESSION 2022/2023		
I declare that this thesis is classi	fied as:	
CONFIDENTIAL (Contains confidential information under the Official		
□ RESTRICTED	(Contains restricted information as specified by the	
<ul> <li>☑ OPEN ACCESS</li> <li>☑ OPEN ACCESS</li> <li>I agree that my thesis to be published as online open accer</li> <li>(Full Text)</li> </ul>		
<ol> <li>I acknowledge that Universiti Malaysia Pahang reserves the following rights:</li> <li>The Thesis is the Property of Universiti Malaysia Pahang</li> <li>The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.</li> <li>The Library has the right to make copies of the thesis for academic exchange.</li> </ol>		
Certified by:		
(Student's Signature)		
New IC/Passport Number Date: 08/02/2023	Dr. Zafril Rizal Bin M Azmi Name of Supervisor Date: 08/02/2023	

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

#### THESIS DECLARATION LETTER (OPTIONAL)

Librarian, *Perpustakaan Universiti Malaysia Pahang*, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300, Gambang, Kuantan.

Dear Sir,

#### CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name<br/>Thesis TitleReasons(i)(ii)(iii)(iii)(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



## SUPERVISOR'S DECLARATION

I/<del>We\*</del> hereby declare that I/<del>We\*</del> have checked this thesis/project\* and in my/<del>our\*</del> opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) With Honours.

(Supervisor's Signature) Full Name : Dr. Zafril Rizal Bin M Azmi Position : Senior Lecturer Date : 08/02/2023



# **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

# t

(Student's Signature) Full Name : LIM KAH CHUN ID Number : CB19037 Date : 08/02/2023 **UMP** Confession Application

# LIM KAH CHUN

Thesis submitted in fulfillment of the requirements for the award of the degree of Bachelor of Computer Science (SOFTWARE ENGINEERING)

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

#### ACKNOWLEDGEMENTS

First, I would like to say thank you to Dr. Zafril Rizal Bin M Azmi who accept me to become my Final Year Project's supervisor. Dr. Zafril Rizal Bin M Azmi give me a lot of guidance and always give me feedback during the full development process of my Final Year Project entitled "UMP Confession Application".

Next, I would like to thanks to my parent who always support me in physical and mental. They bought me a new laptop immediately without any consider when my laptop is malfunction and having damage problem so I can complete my Final Year Project on time.

Last but not least, I also want to thanks to all my dear friends and my roommate who always give me advice, motivations and positive energy to me while the development is ongoing.

#### ABSTRAK

Pada masa terkini, aplikasi media sosial adalah aplikasi yang paling banyak digunakan di seluruh dunia. Semua orang mesti akan mengguna Facebook, Twitter atau Instagram dalam kehidupan seharian mereka dan ia sudah menjadi sebahagian daripada kehidupan kita sekarang. Aplikasi media sosial boleh membantu orang ramai membina hubungan antara satu sama lain. Media sosial akan menghubungkan orang dengan orang. Ia menyediakan platform untuk orang ramai bercakap apa yang ada dalam hati mereka. Oleh itu, menyediakan aplikasi media sosial yang hanya khusus dan digunakan oleh masyarakat di sesebuah university adalah suatu idea yang menarik. Sebagai contoh, Universiti Malaysia Pahang dipilih sebagai komuniti yang akan memiliki aplikasi media sosial ini. Tajuk aplikasi media sosial ini ialah "UMP Confession Application" dan ia merujuk kepada public page dalam Facebook yang merupakan laman umum untuk tujuan tertentu, dicipta khas untuk pelajar Universiti Malaysia Pahang. Ia adalah platform di mana mereka akan dapat menjerit apa yang tersimpan di dalam diri mereka. Aplikasi ini boleh dibangunkan untuk menggantikannya sebagai platform terbaru untuk pelajar di Universiti Malaysia Pahang.Aplikasi ini melaksanakan metodologi tangkas kerana ia memberikan responsif yang lebih tinggi dan meminimumkan ralat semasa proses pembangunan. Ia terdiri daripada enam fasa iaitu merancang, mereka bentuk, membangunkan, menguji, mengeluarkan dan maklum balas. Pertama, keperluan dan objektif dikenal pasti kemudian teruskan dengan mereka bentuk seni bina sistem aplikasi. Seterusnya membangunkan sistem. Selepas fasa pembangunan, fasa ujian akan menggunakan Ujian Penerimaan Pengguna untuk mengesan kecacatan dan ralat dalam sistem. Aplikasi itu kemudiannya akan dikeluarkan dan mengumpul maklum balas daripada pengguna.

#### ABSTRACT

Nowadays, social media application is the most used application around the world. Everyone will use Facebook, Twitter or Instagram in their daily life and it is already part of our life now. Social media application can help people to build relationship with each other. Social media is about connecting people to people. It provides a platform for people to speak up what inside them. Therefore, it is a interesting idea if there also have an social media application that only specific and used by the community in a university. For example, UMP is chosen as the community that will have this social media application. The title of this social media application is UMP Confession Application and it is reference to public page in Facebook which is a public page for a cause, specially created for University Malaysia Pahang's students. It is a platform where they will be able to shout out what's kept inside them. This application can be develop can replace it as a new platform for the student in UMP. The application implements agile methodology as it provides higher responsiveness and minimize errors during development process. It consists of six phase which is plan, design, develop, test, release and feedback. First, requirements and objective are identified then continue with designing the system architecture of application. Next is developing the system. After the development phase, testing phase will use User Acceptance Test to detect defects and errors in the system. The application will then be released and collect feedback from the users.

# TABLE OF CONTENT

DEC	CLARATION	
TIT	LE PAGE	
ACK	KNOWLEDGEMENTS	ii
ABS	TRAK	iii
ABS	TRACT	iv
TAB	BLE OF CONTENT	v
LIST	Γ OF TABLES	viii
LIST	Γ OF FIGURES	ix
LIST	Г OF SYMBOLS	xii
LIST	Γ OF ABBREVIATIONS	xiii
CHA	APTER 1 INTRODUCTION	14
1.1	Introduction	14
1.2	Problem statement	16
1.3	Objective	17
1.4	Scope	17
1.5	Significance of project	18
1.6	Report organization	19
CHA	APTER 2 LITERATURE REVIEW	20
2.1	Introduction	20
2.2	Existing Application	20
	2.2.1 Anonymous Social Secret Confessions & Chats	21

CHAI	PTER 4	L	107
3.7	Potent	ial Use of Proposed Solution	106
3.6	Testin	g plan	105
	3.5.1	Design Prototype	82
3.5	Proof	of Initial Concept	82
	3.4.2	Database Dictionary	78
	3.4.1	Entity relationship diagram	77
3.4	Data Design		77
	3.3.7	Activity diagram	75
	3.3.6	Use Case Diagram & description	49
	3.3.5	Context diagram	48
	3.3.4	Limitations	47
	3.3.3	Constraints	47
	3.3.2	Non-functional requirements	46
	3.3.1	Functional requirements	44
3.3	Projec	t requirement and proposed design	44
3.2	Projec	t management framework	40
3.1	Introd	uction	40
СНАН	PTER 3	METHODOLOGY	40
2.4	Summ	ary	39
2.3	Comp	arison of Existing Application and the Proposed Application	35
	2.2.3	Speak up	32
	2.2.2	Whisper	27

IMPLEMENTATION, RESULT AND DISCUSSION	107

4	<b>I</b> .1	Introduction	107
4	1.2	Implementation Process	107
4	1.3	Testing and Result Discussion	147
СНАРТ	TER 5		148
CONCI	LUSIC	DN	148
5	5.1	Introduction	148
5	5.2	Objective Revisited	148
5	5.3	Limitation	149
5	5.4	Future Work	149
REFER	ENCI	ES	150
APPENDIX A SRS 15		151	
APPENDIX B SDD			

APPENDIX C UAT

# LIST OF TABLES

Table 2. 1: Comparison of three existing application		36
Table 2. 2:	Advantage and disadvantage of three existing application	38

Table 3. 3. 1: Use case description of manage registration	54
Table 3. 3. 2: Use case description of manage post	58
Table 3. 3. 3: Use case description of manage comment	61
Table 3. 3. 4: Use case description of manage Announcement	64
Table 3. 3. 5: Use case description of manage order type	67
Table 3. 3. 6: Use case description of manage order	71
Table 3. 3. 7: Use case description for manage complaint	74

Table 3. 4. 1: Data Dictionary for User	78
Table 3. 4. 2: Data Dictionary for Admin	78
Table 3. 4. 3: Data Dictionary for Post	79
Table 3. 4. 4: Data Dictionary for Comment	79
Table 3. 4. 5: Data Dictionary for Announcement	80
Table 3. 4. 6: Data Dictionary for Order	81
Table 3. 4. 7: Data Dictionary for Order type	81
Table 3. 4. 8: Data Dictionary for Complaint	81

# LIST OF FIGURES

Figure 2. 2. 1. 1 Logo of Anonymous Social Secret Confessions & Chats application.	21
Figure 2. 2. 1. 2: Login feature of Anonymous Social Secret Confessions & Chats	22
Figure 2. 2. 1. 3:Register Page of Anonymous Social Secret Confessions & Chats	23
Figure 2. 2. 1. 4: Register Page for select face image of Anonymous Social Secret Confessions & Chats	24
Figure 2. 2. 1. 5: Select interest category feature in Anonymous Social Secret Confessions & Chats	25
Figure 2. 2. 1. 6: Main page in Anonymous Social Secret Confessions & Chats	26
Figure 2. 2. 1. 7: Post confession feature in Anonymous Social Secret Confessions & Chats	26
Figure 2. 2. 2. 1: Logo of Whisper	27
Figure 2. 2. 2. 2:Main Page of Whisper	28
Figure 2. 2. 2. 3: Post message with image on Whisper	29
Figure 2. 2. 2. 4: View post in nearby feature on Whisper	30
Figure 2. 2. 2. 5: Create group feature on Whisper	30
Figure 2. 2. 2. 6: Group feature on Whisper	31
Eigure 2, 2, 2, 1, Logo of Speak up	22
Figure 2. 2. 3. 1: Logo of Speak up	32 22
Figure 2. 2. 3. 2: Kegister feature in Speak up	32 22
Figure 2. 2. 3. 5: Select category for company in Speak up	33 24
Figure 2. 2. 5. 4. write comment page in Speak up	34
Figure 3. 2. 1: Process of Agile development	41
Figure 3. 3. 1: Context diagram for UMP Confession Application	48
Figure 3. 3. 2: Interaction between UMP Confession application and the actors	49
Figure 3. 3. 3 : Use case for manage registration	50
Figure 3. 3. 4: Use case for manage post	54
Figure 3. 3. 5: Use case for manage comment	59
Figure 3. 3. 6: Use case for manage Announcement	62
Figure 3. 3. 7: Use case for manage order type	65
Figure 3. 3. 8: Use case for make order	68

Figure 3. 3. 9 : Use case for manage complaint	72
Figure 3. 3. 10: Activity diagram of user	75
Figure 3. 3. 11: Activity diagram of admin	76
Figure 3. 4. 1: Entity relationship diagram of UMP Confession Application	77
Figure 3. 5. 1: Login interface of UMP Confession Application	82
Figure 3. 5. 2: LoginAsAdmin interface of UMP Confession Application	83
Figure 3. 5. 3: Register interface of UMP Confession Application	84
Figure 3. 5. 4: UserPage interface of UMP Confession Application	85
Figure 3. 5. 5: AdminPage interface of UMP Confession Application	86
Figure 3. 5. 6: CheckAccount interface of UMP Confession Application	87
Figure 3. 5. 7: ManagePost interface of UMP Confession Application	88
Figure 3. 5. 8: MakePost interface of UMP Confession Application	89
Figure 3. 5. 9: EditPost interface of UMP Confession Application	90
Figure 3. 5. 10: ViewPost interface of UMP Confession Application	91
Figure 3. 5. 11: ManageOrder interface of UMP Confession Application	92
Figure 3. 5. 12 : AddOrder interface of UMP Confession Application	93
Figure 3. 5. 13: ViewOrder interface of UMP Confession Application	94
Figure 3. 5. 14: ManageAnnouncement interface of UMP Confession Application	95
Figure 3. 5. 15: MakeAnnouncement interface of UMP Confession Application	96
Figure 3. 5. 16: EditAnnouncement interface of UMP Confession Application	97
Figure 3. 5. 17: ManageOrderType interface of UMP Confession Application	98
Figure 3. 5. 18: AddOrderType interface of UMP Confession Application	99
Figure 3. 5. 19: AdminManagePost interface of UMP Confession Application	100
Figure 3. 5. 20: My Complaint interface of UMP Confession Application	101
Figure 3. 5. 21: MakeComplaint interface of UMP Confession Application	102
Figure 3. 5. 22: Manage Complaint interface of UMP Confession Application	103
Figure 3. 5. 23: add comment interface of UMP Confession Application	104

Figure 4. 1: Interface of Main Page of Application	108
Figure 4. 2: Interface of register page	109
Figure 4. 3: Interface of user main page	110
Figure 4. 4: Interface of make post page	111
Figure 4. 5: Interface of my post page	112
Figure 4. 6: Interface of view post page	113
Figure 4. 7: Interface of view post by using filter	114
Figure 4. 8: Interface of my complaint	115
Figure 4. 9: Interface of my complaint	116
Figure 4. 10: Interface of make complaint	117
Figure 4. 11: Interface of View post (On single post)	118
Figure 4. 12: Interface of View comments	119
Figure 4. 13: Interface of my order	120
Figure 4. 14: Interface of make order	121
Figure 4. 15: Interface of make order	122
Figure 4. 16: Interface of view order	123
Figure 4. 17: Interface of insert order type ID	124
Figure 4. 18: Interface of manage order	125
Figure 4. 19: Interface of manage order	126
Figure 4. 20: Interface of login page as admin	127
Figure 4. 21: Interface of admin main page	128
Figure 4. 22: Interface of manage announcement	129
Figure 4. 23: Interface of make announcement	130
Figure 4. 24: Interface of edit announcement	131
Figure 4. 25: Interface of admin manage post	132
Figure 4. 26: Interface of check account page	133
Figure 4. 27: Interface of check account (Approve account)	134
Figure 4. 28: Interface of manage order type	135
Figure 4. 29: Interface of add new order type	136
Figure 4. 30: Interface of admin manage complaint	137
Figure 4. 31: Interface of admin manage complaint (Give reply to user)	138
Figure 4. 32: Android Studio	146
Figure 4. 33: Real-time database for the application	146

# LIST OF SYMBOLS

# LIST OF ABBREVIATIONS

UMP	Universiti Malaysia Pahang
SRS	Software Requirement Specification
SDD	Software Design Document
UAT	User Acceptance Test
UMPCA	UMP Confession Application
ERD	Entity relationship diagram

#### **CHAPTER 1**

#### **INTRODUCTION**

#### 1.1 Introduction

According to the news from internet, there are a lot of students always having problem and facing the issue which is about carrying too much stress in their life. Not only because of their study life and may be related to their family or friend. It may happen due to the student cannot find out any platform or method to speak out their stress and reduce their stress. Last year (2021), the social were saddened by the news of the demise of two UiTM students. These 2 cases is happening on the same month which is July. Both of them pass away due to stress from Studying, both of them are working late on their assignment and coursework on the day that they pass away. (*UITM students die due to stress from studying: What is the Mohe doing to address this issue*)

After these case happening, it can be state that carry too much stress can have a very serious consequences, even though sometimes some people consider that a little stress is not a big issue but these pressures will slowly build up inside us and become a timer bomb that may cause a big explosion inside us. Therefore, everyone should vent out their emotions appropriately from time to time. Whether its sadness, anger, or any of these negative emotion that should be released to avoid bad things happen. Although speaking it out may not really can eliminate all the pressure, but it is better than suppressing all this negative energy inside us.

In fact, there are some university seems to have a Facebook Page which is related to confession. Many people may don't think so much about these Facebook page at first but as the times pass, they will found out that there are very interesting and also can be a really helpful things for the students. It give a big chance and suitable platform to the student for speaking out that anything that inside them. In Facebook, there are also exist 2 Facebook page which is UMP Confession but it is not really an official Facebook page in this university. (*Ump confessions*) It is just a Facebook page that allow the student to speak or shout out what is keeping inside them without other people know their identity. The people who want to post their confession can submit their confession with or without picture by google form and the admin of that Facebook Page will post it at that Facebook Page. One of them is already stop for using while the other is still available. It receive almost 10 thousand likes by the people and most of them may be the student who study at UMP at past or present.

This confession platform is getting a lot of response from the students according to this information. What if this platform can become an official application which can used by the staff of UMP and also the student in UMP? It is going to be a challenging project on how can the developer make this application with more function and make everyone in UMP can enjoy to use it. This confession platform is having a good potential to be a useful and popular application in UMP.

Since this is just a confession platform that belong to Facebook, the idea of developer is to make an application to replace this UMP Confession in Facebook page and more function will be implement to make that application to be more functional and become a useful application for the student and staff of UMP.

#### **1.2 Problem statement**

There is a public page on Facebook for student in UMP to let them post their confession without let anyone know the confession is belong to whom. If they continue to use this platform to make confessions, there will actually be some problems.

First, there is a third party in this platform and that is Facebook, so it will be more convenient and safe to have a separate database and create a separate application to become a new platform for the student in UMP to make confession. In addition, since this platform is not under the management of UMP, it will be more convenient if UMP could create a new application which is managed by UMP itself. This type of platform is popular among the student. It will be a successful application if UMP management can make this application be official for collective good.

In addition, if the app is only used for confession, it will lack of use and not reality. Other features such as make complaint and view announcement that post by staff can be implement into the application to encourage more students to download and use this application. Besides, more function can be add into this application if the situation and condition is allow. For example, order food function which the developer make collaboration with the owner of cafeteria in UMP.

A bunch of confession posts in Facebook Page will look very messy. If the system can divide the types of posts to distinguish each confession, it will be more convenient for the users to view the content that they want to see according to their interest, for example, the confession can be classified into different type of post such as: study, entertainment, request for help, resale item, sports, etc.

## 1.3 Objective

- i. To study the existing application related to speak up what inside the user by posting message on a platform.
- ii. To develop a social media application using Android studio and become a platform for the student in UMP to speak out their emotion and chat with others that may have same interest so they can reduce their stress.
- iii. To evaluate the functionality of UMP Confession application.

## 1.4 Scope

Scope of project defines the focus area of the subject matter. Must be related to the project.

User scope:

- 1. Student who study in UMP
- 2. Staff who work in UMP
- 3. Anyone who stay in area of UMP such as operator of cafeteria in UMP

#### **1.5** Significance of project

#### 1. Student who study in UMP

To develop an application that can replace the Facebook Page for confession and make it more functional and useful to give more benefit to the student in UMP. This project is not beneficial to entire society directly but it is beneficial to the student in UMP. Besides, it can also be a template of application that used by other university so they can have same type of application to use in their specific university.

#### 2. Staff who work in UMP

In the plan for the development of this application, it will add some function such as make complaint and make an official announcement to student. The student can make complaint by using this application and the staff can use this application to check these complaint and make improvement or give response back to the student. It may not an official way to make complaint but sometimes the student will refuse to make complaint if things need to be official. The staff can choose to handle these complaint directly or tell them to give more details in official way such as using ECOM. Besides, staff also can use this application to post any official announcement because if the application is success to make more student use and open it daily just like everyone open their social media application (Facebook. Instagram, Twitter), it will be easier for the staff to let more student know the announcement.

#### 3. Anyone who stay in area of UMP such as operator of cafeteria in UMP

In the early plan, the application will also have the function such as make order and make booking. For example, the operator of cafeteria can make cooperation with this application so the student can make pre-order by this application so the student can take their food faster when they came to cafeteria. The customer who will buy food from this cafeteria may increase since the student can take their food faster with pre order. The staff of any facilities that may need booking before using the facilities such as gym can also use this application to receive booking from the students. It will make them manage their facilities easier.

#### **1.6** Report organization

This thesis consists of five chapters. Chapter one discussed introduction of my project which is UMP confession application. This chapter consists of problem statement, objective and scope of project. The literature review is discussed in chapter 2. This chapter will discuss three existing applications which is also used to speak up what inside the user and comparison between them will be conduct.

Chapter 3 will talk about the methodology for doing this project. This chapter will describe on applied project management framework and requirement in the project. Chapter 4 will cover the outcome for this project. This chapter will describe the implemented works and results for this project. Chapter 5 is last chapter and it will make a conclusion on this project.

#### **CHAPTER 2**

#### LITERATURE REVIEW

#### 2.1 Introduction

In this chapter, three relevant existing applications will going be choose and each of them will be compare and make a conclusion to identify how the proposed application can be done more perfect. These three existing applications will be reviewed, explained, discussed and compared. The advantages, disadvantages and comparisons of the existing applications will also be discussed based on their features, function, platform, and other relevant aspects. The aims of this chapter is to give a better view and understanding of how the proposed application should be able to conduct.

#### 2.2 Existing Application

The chosen existing applications will be explained, analysed and discussed in this section. The discussion aims to have better enhancement and avoid any possible error or issue happen in my proposed application by going through these applications to observe the advantages and the disadvantages of these applications. My propose application is type of application which is focus on communication and about posting the confession of the user. After doing a research on internet and Google Play store, the applications that will be choose are "Anonymous Social Secret Confessions & Chats", "Whisper" and "Speak up". (Mrchawla, *5 apps that let you express what you feel, anonymously #section66a* 2015) These application are included in type of applications which will allow the user to speak up or post anything to a platform. These application also can be state as type of Social Media Application

The term "Social Media Application" refers to the range of web applications based on Web 2.0 technologies that will allow their users to socially interact with one another through online. These applications allow them to gather, represent, process, use, and disseminate information online in diverse ways and through a variety of media, producing dynamic virtual spaces. The case of my propose application is similar to these type of application. These type of application will provide the users to have a platform to post their confession and let other people to view their confession post or give them some positive comment. (X., *Handbook of Research on Learning Outcomes and opportunities in the Digital age* 2016)

To make better analysis and investigate on these application, these application will be download and install on phone for testing their function and features.

#### 2.2.1 Anonymous Social Secret Confessions & Chats



Figure 2. 2. 1. 1 Logo of Anonymous Social Secret Confessions & Chats application.

Anonymous Social Secret Confessions & Chats is the only application which produce by Telling ly company. This application is widely used by the user who want to speak up the secret or confessions inside them. This application provides the user a platform where they are free to share their secrets and stories without letting other people about themselves. The post and all the comment will be anonymous to other user. Everyone or anyone who are shy in reality can talk to a stranger and reply to their advice without worry anything since they will never know who they are. (*Anonymous Social Secret Confessions & Chats - Apps on Google Play*)



Figure 2. 2. 1. 2: Login feature of Anonymous Social Secret Confessions & Chats

This application will require their user to make an account before using the application It only need user to input name, username, password age, and gender for register. However, it doesn't have any verification process so it will be a simple and easy process to register an account. Telling.ly company did not ask for any email or phone number so the user need to keep their password safe and remember their password since there are no others way for the user to find their password.



Figure 2. 2. 1. 3:Register Page of Anonymous Social Secret Confessions & Chats

When the user select their age in register process, different comment will given by the system according to different age. After register the account, many picture of face draw in cartoon style will appear and ask user to choose one to become their profile picture since this application is going to be anonymous. These mini but creative detail will make the user feel the application is more interesting and have will to keep using the application.



Figure 2. 2. 1. 4: Register Page for select face image of Anonymous Social Secret Confessions & Chats

The usage of select category for user after register an account is really helpful for the user because it will make the user can find the post that related to a specific topic that they are interest easily. For different category, a "#" to the beginning of an unbroken word or phrase will create a hashtag. When the user use a hashtag in their post, it will becomes linked to all of the other post that include this hashtag. It will allow the user to easily follow the topics that they're interested in. For example, alone, anger, confused and family. If the user can select any category as their interest, the content or post which is related to this category will be display to the user.



Figure 2. 2. 1. 5: Select interest category feature in Anonymous Social Secret Confessions & Chats

To create a post about the confession, the user can click the button on top left and fill in the details of their confession to post it to public and let other people read or comment. The step is simple and not going to consume too much time. The user only need to give title, content, and the user can give tag on their post to make hashtag and linked to this category of post. The user also can pick an picture which provide by the system and that picture is related to the category that the user add for the post.



Figure 2. 2. 1. 6: Main page in Anonymous Social Secret Confessions & Chats

Write Your Confession					×	
Note: Your post will only visible in your timeline if it match with your interest. You can see all your post in My post section.						
litie						
i am doing my psm						
Content						
need to work hard for my psm						
					~	
Add Tags X						
#ADDICTION		#ALONE		#ANG	#ANGER	
#CHILDHOOD		#CONFUSED		#D	#DESPAIR	
#FAMILY	#FEAR		#FRIENDS	5	#FUN	
#GUILT	#HATE		#JEALOUS	Y	#LOVE	
#REGRET	#RELATIONSHIPS		#RI	#REVENGE		
#SEX	#SPOUSE #WEIRD		#WORK			

Figure 2. 2. 1. 7: Post confession feature in Anonymous Social Secret Confessions & Chats

#### 2.2.2 Whisper



Figure 2. 2. 2. 1: Logo of Whisper

Whisper is an online community where millions of people around the world share their real thoughts, trade advice with each other, and get the inside scoop.. This application launched by the MediaLab.AI, Inc. Unlike most of the social media application that available in the market, this application is in form of anonymous social media, which will allow their users to post and share photo or video messages anonymously. (*Whisper*)



Figure 2. 2. 2. 2: Main Page of Whisper

Different to application in 2.2.1, register and login feature are not necessary for Whisper. The user who install this application can be use directly without any account. After open the application, the user can click the + button below to make a post by typing their confession and the system will automatic suggest a background image for them. If the background image given by system is not the user favourite ,the user also can search for other image by input the keyword for the image that the user wish to use as background. The user also can choose to use their own picture or take any picture at that moment. Different font also can be use which provide by the system to customize the post of user.



Figure 2. 2. 2. 3: Post message with image on Whisper

Besides, it can allow user to view the post in different way. They can choose to view the posts which is popular, nearby or latest. For popular, those post which are more view or receive more like and share by other user will be display to the user. For nearby, the user can set whether they want to see the post that make by the user closer than or more farther than themselves in real world. This option is not really suitable for this application because the main element of this application is anonymous. It will be pointless if the location of the user will be identify by other user. For latest, the user can view all the post that just make by other user recently.

Besides, group function also had implement into this application. The user are freely to create a group and set a theme for that group so the other user who interest in that specific theme can ask for join the group to chat and discuss together in anonymous way.



Figure 2. 2. 2. 4: View post in nearby feature on Whisper



Figure 2. 2. 2. 5: Create group feature on Whisper



Figure 2. 2. 2. 6: Group feature on Whisper
### 2.2.3 Speak up



Figure 2. 2. 3. 1: Logo of Speak up

Speak Up is a mobile application where consumers can voice their complaints or provide suggestions and feedback for a company, product or service. The aim of this application is to revolutionize the age-old complaint redressal system by leveraging Social media which is now rapidly evolving as a one-stop response medium for organizations. This is not include in social media application but it also type of application which allow the user to speak up what inside them and point up their opinion or complaint to certain things. (*Speakup | Consumer Complaints - apps on Google Play*)



Figure 2. 2. 3. 2:Register feature in Speak up

The user need to login by using their Google Account before using the application. In the main page of Speak up, there are a lot of icon which represent different section or category of product or services. The user can just find the company that they wish to give comment by select the category of that company. For example, there are flight& Airports, Restaurants & Cafes, and Banking. Since this application is design for people who stay in American so most of the product and services is not familiar to us as Malaysian.



Figure 2. 2. 3. 3: Select category for company in Speak up

This application is more on complaint use but it will not limited their user, Speak Up also can be register for sending feedback or give a word of appreciation to the organizations. When the user choose which company that they want to make comment on it, they can choose which type of content that wish to send such as complaint, feedback and compliment. Next they can write the message in textbox and they can prefer to add photo or record their voice for more elaboration. Their message will get directed automatically to organizations' multiple channels like E-mail, Twitter etc.



Figure 2. 2. 3. 4: Write comment page in Speak up

However, it will ask the user to verify by using phone number if they want to submit the comment. During the verification process by phone number, it said it is invalid phone number and it may because of this application only available for the user who stay in American. This issue will make the people who don't have phone number with valid format cannot use this application.

## 2.3 Comparison of Existing Application and the Proposed Application

In this part, a comparison table will be created based on different aspects to compare these existing applications in the Table 2.1. In Table 2.2, the advantage and disadvantages of these existing applications will be discussed.

Specification	Anonymous Social	Whisper	Speak up
	Secret Confessions &		
	Chats		
Platform	Android, IOS	Android, IOS	Android, IOS
Application size	21MB	27MB	7.7MB
Application	Social	Social	Communication
category			
User Category	Rated for 18+	Rated for 12+	Everyone
Rating on Play	3.4/5.0 with 244 review	3.4/5.0 with	No review
Store (Data		236,105 review	
collected on April			
2022)			
Downloads on	10,000+	10,000,000+	100,000+
Play Store (Data			
collected on April			
2022)			
Main use	Post confession	Post confession	Make complaint

Register and login	Yes	No	Yes
Set Category for	Yes and up to 15	No	Yes, it can give
post message	categories		message for
			complaint,
			feedback,
			compliment
Social usage	Yes	Yes	No
Anonymous	Yes	Yes	No
Check history for	Yes	Yes	Yes
posting			
View post in	Yes	Yes	No
different way			
Post with	Can choose picture that	Can post	Can post with
Multimedia	given by system only	with user own	picture and audio
		picture	file

Table 2. 1: Comparison of three existing application

Application	Advantage	Disadvantages
Anonymous	- More secure because user need to	-User cannot choose their
Social Secret	register first with password	own picture to post
Confessions &		
Chats	- User can set category for their	-Only can view post for
	post so the other user can view post	different category but cannot
	according to the category that they	view the post according
	interest only	suitable way such as latest or
		popular
	-Can post in 4 different way which	
	is confession, feeling, story and ask	
	question	
	4	
	The III look more interacting	
	- The OT look more interesting	
Whisper	-User can choose their own picture	-The user being force to
	as background and use different	write message with an image
	font to customize their post	but the message of user may
		become difficult to read
	-Create and join group feature	and length of text message of
	allow the user who have same	user may being limit
	interest to discuss and chat to each	
	other	-No account need and it will
		be not secure and if the
	-Allow the user to view the post in	application is deleted, the
	different way such as popular,	user cannot find their history
	latest and nearby.	record anymore
	•	

	-The user don't need to register and	-Nearby feature will expose
	login by any account since all the	user location in real world
	post will be anonymous	
C L L	TT	
<b>Speak</b> Up	- User can give comment to a	-Availability for this
	specific organization easier	application is not really good
	because there are a lot of	since it only limit-use by the
	organization that available in the	user who stay in America
	system for give comment	only
	-All the organization in group into	-There exist a lot of
	different category orderly and the	organization but most of
	user can find the organization that	them is only organization
	they wish to give comment faster	which belong to American
		and not widely used by
	-The user can use one application	people in other country
	only to make different comment to	
	different organization.	-Only use for speak up about
		complaint, feedback and
		compliment to an
		organization but not use for
		personal issue

Table 2. 2: Advantage and disadvantage of three existing application

### 2.4 Summary

Based on my analysis and comparison that done among these three existing application, each application has its unique part and some of them may exist some issue. For Anonymous Social Secret Confessions & Chats and Whisper, it more similar to the proposed application since the main function of these two application is post confession and speak up what inside them without expose their real identity. For Speak Up, it is an application which is more focus on make complaint or give feedback or compliment to an organization but it also included to being investigate because the proposed application will implement function for make complaint.

From the studies have been done, the proposed application will try to implement those features that make advantage of this application and try to avoid having the issue that make disadvantage of this application. First, the application must be anonymous since sometimes we don't want to let other know when we are using social media. Anonymous Social Secret Confessions is also fully anonymous while we are using it to post confession and our proposed application will implement anonymous way for user to post their confession. The other user who view the post will never know the person who make that post for privacy of the user.

For Whisper, there are a feature will allow user to join group so they can find other user who have same interest to discuss and chat to each other. Anonymous Social Secret Confessions also can view post in different category so the proposed application also can set their post as different type such as Music, Game, Movie and more and user can view different post by a filter. For Speak Up, complaint type are too many and it will confuse user so proposed application will not having categories for complaint. User just need to write anything about their complaint in short.

### **CHAPTER 3**

#### METHODOLOGY

#### 3.1 Introduction

In this chapter, the methodology in developing this UMP confessions application will be discussed as well as its design, interface and the framework architecture. The requirements will be write down in order and list in a table. As a UMP students which is also the potential user of proposed application, the requirements can be define by developer himself. Further details of the application development will be stated in the other document which is Software Requirement Specification (SRS) and Software Design Description (SDD) documents.

### **3.2 Project management framework**

This section will describe on applied or used project management framework in the project which is agile methodology. Agile methodology is an approach that emphasize on more flexible planning and development. It is incremental and iterative which allows modification to be implement in more flexible way. If compared to most of the framework that common used like waterfall development model, it is not easy to make modification during the development process because waterfall model is a linear and sequential approach. The whole development process will be more flexible to make changes by using agile methodology because changes can be implemented easier compare to other framework and can change according to adapt the customer's requirement which offers higher adaptability. (*Agile Software Development Lifecycle Phases explained* 2022)

The advantages of using Agile are higher work transparency which allows the stakeholders know about the development progress and give some opinion for improvement during the development process. It can also minimizes errors during the development as constant interaction is practiced constantly. Stakeholders that involved in agile methodology has high satisfaction towards the system. Besides, this framework will promote effective communications as this method practices constant interactions between developers and the user, the user can change the requirements and needs even at last stage.



Figure 3. 2. 1: Process of Agile development

#### Phase 1: Plan

Before start to design the proposed application for this project, a document must be prepared which is software requirement specification (SRS) which included the outcome such as expected interface design of the project and the functionality will be provide. The requirements for proposed application will be created in this phase.

### Phase 2: Design

This phase will define the programming language and database that will be used in the project. The programming language will be use in this project is JAVA. At the same stage, prototype of the proposed application will be create base on the expected user interface of the application.

#### Phase 3: Develop

This development phase includes coding and translating software design documentation into actual applications. This is the most important stage of SDLC because it will form the foundation of entire process. In this phase, interfaces and functions of proposed application that had already plan and design in previous phases will be implement.

### Phase 4: Test

In this phase, developer will ensure that the proposed application can be use without error and compatible with everything else that has been built in previous phase. Many times of tests will be done to check wheatear the code is without any error and all requirement have been meet in this phase.

### **Phase 5: Release**

The application is now fully deployed and can release to the user. This action puts him in the maintenance phase. During this phase, developer will provide ongoing support to keep the system of the application can run smoothly on Android smartphone and fix any new bugs if exist. Over time, further iterations are possible to be update an existing application or add other functionality for better user experience.

## Phase 6: Feedback

This phase will discuss the progress had made toward meeting the requirements after all previous phases have been completed. The developer will present the ideas for correcting the issues that appeared during the previous phases. After that, the software development lifecycle phases will restart with new iteration.

### 3.3 **Project requirement and proposed design**

This section for project requirement will describe all the requirement related to the proposed project include Functional and non-functional requirements, Constraints and limitations. For proposed design, it will describe the proposed design which related to project requirement such as Context Diagram, Use Case Diagram & description and Activity diagram for the application.

#### **3.3.1** Functional requirements

Functional requirements are the product features or functions that developers must implement to enable users to accomplish their tasks. It is important to make them clear both for the development team and the stakeholders. In General, functional requirements will describe the system behaviour under specific conditions. The list below show the functional requirements of UMP Confession Application.

1. The system will allow user to register an account by upload an image to prove that they are related to UMP for the verification.

2. The system ask user to login with their id and valid password.

3. The system sends information of user to admin for manage.

4. The system can allow admin to delete user if admin found out that the user who make register did not have image to prove they are from UMP.

5. The system can allow admin to approve user account as valid to login if admin found out that the user who make register is valid to use the application.

6. The system will display all post to the user when they login successfully.

7. The system will display announcement that created by admin to all user at the top of before display post.

8. The system will give different choice for user to choose as the category of their post.

9. The system will post directly after user input the form for post.

10. The system allow the user to delete or manage their own post.

11. The system will allow the user to add comment on other post.

12. The system will allow admin to manage all the post that created by all user

13. The system will allow admin to add a new order type for user to choose as a order type of their order.

14. The system allows admin to delete order type.

15. The system will allow the user to add comment on the order to update status of order.

16. The system will allow the user to manage their own order.

17. The system will allow user to choose for different function at user main page.

18. The system will allow admin to choose for different function at user main page.

19. The system will allow user to view post in a specific categories by select type of post.

20. The system will allow user to make complaint and delete their own complaint.

21. The system will allow user to view status of their complaint and reply from admin.

22. The system will allow admin to reply on complaint that user made.

### 3.3.2 Non-functional requirements

For non-functional requirements, there are not related to system functionality and they rather define how the system should perform. List below show the non-functional requirement of UMP confession application. (Ng, *Non-functional requirement of the mobile development system*. 2019)

- 1. Security: The system only allow user to log in success with a valid password.
- 2. Performance: The time for the system to make post be visible to other users after user submit the form should not take more than 3 second.
- 3. Usability: The user interface of the application should be easier to understand and use by the user.
- 4. Reliability: Any important action such as create post should be acknowledged with confirmation by the user.
- 5. Availability: The user can use this application at any time and any place when having internet connection.
- 6. Scalability: When the user data increases, the application should be capable of handling these huge data without any error and delay.

### 3.3.3 Constraints

-The user who register account must submit register with a photo as proof for admin to verify their account is valid to use the application or not.

-No hate or any disrespect word allow in this application.

-The user must register and login first before using this application.

-The application only can be used with the access of network.

## 3.3.4 Limitations

-The application is only available for the user who use Android as the operating system of their smart phone

-The application only available for the user who are from Universiti Malaysia Pahang.

-Minimum SDK is API 21: Android 5.0 (Lollipop) and the application will run on approximately 98 % of devices.

### 3.3.5 Context diagram



Figure 3. 3. 1: Context diagram for UMP Confession Application

Figure 3.3.1 shows the context diagram of the UMP Confession Application. Context diagram refers to DFD (data flow diagram) that makes details and boundaries of the designed system clearer that it makes project's scope will be simpler and straightforward. The context diagram consists of two entities who are user and admin who are going to interact with the application.

First, the user will need to insert user information and password to application so they can have completed register. The user who give correct username and password can have valid login and they can input any content of post or comment to the application and they also can view other post or comment from the application which will retrieve data. The user is also allowed to input order details into application and view order details from application. Beside, user also can view announcement that create by admin. They also can make complaint by input content of complaint and view reply from admin for their complaint.

For admin, they can login by input correct username and password into the application. After login successfully, admin can view all post, user information, announcement and order type from the application. Besides, admin can input new announcement and new order type data into the application. They also can view all complaint from user and reply to them.

## 3.3.6 Use Case Diagram & description



Figure 3. 3. 2: Interaction between UMP Confession application and the actors

# Manage Registration

# Use Case Diagram



Figure 3. 3. 3 : Use case for manage registration

Use Case ID	UMPCA-UC-100
Use Case Name	Manage registration
Brief Description	This use case allow user to register an account and login to use this application. The admin also can approve the user account so they can login into application or reject them.
Actor	User and admin

Pre – Condition	User did not log into their account and
	don't have any account yet.
Basic Flow	User
	1. The use case start when user open the
	UMP Confession Application.
	2. The application will display
	< <login>&gt; interface.</login>
	C C
	3 User click "Register" button and
	application will redirect user to
	application will redirect user to
	< <register>&gt; interface.</register>
	4. User input all the required details of
	user.
	5. User click "Register" button.
	6. The application will display
	< <login>&gt; interface.</login>
Alternative Flow	
Alternative Flow	A1: Having account (User)
	1. User or admin input their username
	and password and click "Login" button.
	2. The application will redirect user to
	< <user page="">&gt; interface if username and</user>
	-

password is valid and they are approve
by the admin to register account.
A2: Check account (View account)
(Admin)
1. Continue step 2 in A1.
2. Admin click "Check account" button
and redirect to < <check account="">&gt;</check>
interface.
3. User account who are not valid will be
display.
1 5
A3: Check account (Approve account)
A3: Check account (Approve account) (Admin)
A3: Check account (Approve account) (Admin)
A3: Check account (Approve account) (Admin)
<ul><li>A3: Check account (Approve account)</li><li>(Admin)</li><li>1. Continue step 3 in A2.</li></ul>
<ul> <li>A3: Check account (Approve account)</li> <li>(Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user account valid to use</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user account valid to use.</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user account valid to use.</li> </ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user account valid to use.</li> <li>4. Application redirect to &lt;<check< li=""> </check<></li></ul>
<ul> <li>A3: Check account (Approve account) (Admin)</li> <li>1. Continue step 3 in A2.</li> <li>2. Admin can click on each user to check their uploaded image to see either they are valid to register account or not.</li> <li>3. Click "Approve" button to set the user account valid to use.</li> <li>4. Application redirect to &lt;<check account&gt;&gt; interface.</check </li> </ul>

	A 4. Login og odmin (Admin)
	A4: Login as aunim (Aunim)
	1. Admin click "Login as admin" button
	to redirect to login as admin interface
	to redirect to login as admin interface.
	2. Admin input username and password
	of admin which is "admin" and
	admin123".
	3. Admin click "Login" button.
	C
	4. The application will redirect user to
	< <admin page="">&gt; interface if username</admin>
	and password is valid.
	-
	A5: Check account (Reject account)
	(Admin)
	1. Continue step 3 in A2.
	2. Admin can click on each user to check
	their uploaded image to see either they
	and upfortated image to see entitier mey
	are value to register account or not.
	3. Click "Reject" button to set the user
	account invalid to use and delete it from
	datahasa
	ualabase.
Exception Flow	E1: Wrong username and password or
	invalid account

	1. User will redirect to < <login>&gt; page.</login>
	2. Use case continue to step 1 in Alternative flow (A1).
Post-Condition	User can register an account which valid to use and login to the application.

Table 3. 3. 1: Use case description of manage registration

# Manage Post

# Use Case Diagram



Figure 3. 3. 4: Use case for manage post

Use Case ID	UMPCA-UC-200
Use Case Name	Manage Post
Brief Description	This use case allow user to create a post for confession or complaint and they can edit or delete it. It also allow admin to delete post.
Actor	User and admin
Pre – Condition	User had register an account and already login to the system.
Basic Flow	<ul> <li>User</li> <li>1. The use case start when user login to the UMP Confession Application.</li> <li>2. The application will display &lt;<user page="">&gt; interface.</user></li> <li>3. User click "View Post" button and system redirect to &lt;<view post="">&gt; interface.</view></li> <li>3. The system will display the post from the database at &lt;<view post="">&gt; interface.</view></li> </ul>

Alternative Flow	A1: Create post (User)
	1. User continue at step 2 of basic flow and click "Make Post" button.
	2. The application will redirect user to < <make post="">&gt; interface.</make>
	3. User input the content of post , select type of post that the user want to choose, and upload image if they want to post together
	4. User click < <done>&gt; button to save data to database and application will redirect to "User Page"</done>
	A2: Edit post (User)
	1. User continue at step 2 of basic flow and click "My Post" button.
	2. The application will redirect user to < <manage post="">&gt; interface.</manage>
	3. User select the post that they want to edit and click "Edit" button to go to < <edit post="">&gt; Interface.</edit>
	4. User edit the content and click "Done" button to update the post.

5. Application will redirect to < <manage page="">&gt;</manage>
interface.
<b>A3.</b> Delete nost (User)
AS. Delete post (User)
1. User continue at step 2 of basic flow and click "My
Post" button.
2. The application will redirect user to < <manage< td=""></manage<>
post>> interface.
3 User select the post that they want to delete and
aliak "Dalate" button
chek Delete buttoli.
4. Application will redirect to << Manage Post>>.
A4: Delete post as admin (Admin)
1. After login as admin, admin can click "Manage
Post" button in < <admin page="">&gt; Interface and</admin>
application will redirect admin to < <admin manage<="" td=""></admin>
nost>> interface
post > interface.
2. Admin select the post that they want to delete and
click delete button.
3. Application will redirect to < <admin page="">&gt;.</admin>
A5: View different type of post (User)

	1. User continue at step 3 of basic flow, the user can click dropdown list menu to select type of post that they want to view.
	2. Click < <search>&gt; button after select the type of post that user want and system will redirect to &lt;<view in="" post="" type="">&gt; interface.</view></search>
	3. System will display all post that the type of post is same as user option.
Post-Condition	The post can be manage by the user and admin successfully.

Table 3. 3. 2: Use case description of manage post

# Manage Comment

Use Case Diagram



Figure 3. 3. 5: Use case for manage comment

Use Case ID	UMPCA-UC-300
Use Case Name	Manage Comment

Brief Description	This use case allow user to manage comment on any post.
Actor	User
Pre – Condition	User had register an account and already login to the system.
Basic Flow	<ol> <li>User</li> <li>The use case start when user login to the UMP Confession Application and system will redirect to &lt;<user page="">&gt; interface.</user></li> <li>User click "View Post" button and system will redirect to &lt;<view post="">&gt; interface.</view></li> <li>The system will display the post from the database.</li> <li>The user choose the post that want to make comment and will redirect to &lt;<view post="">&gt; interface.</view></li> <li>The user insert the content of comment and click "Comment" button.</li> <li>The application will redirect to &lt;&lt; View Post&gt;&gt; interface.</li> </ol>
Alternative Flow	A1 View comment

	1. Continue Step 3 in basic flow.
	2. Click on "View comment" button and system will redirect to << view comment >> interface.
	3. All comment of the selected post will be retrieve
	from database and display at << view comment >> interface.
Post-Condition	The comment will be added below the post.

Table 3. 3. 3: Use case description of manage comment

Manage Announcement

Use Case Diagram





Use Case ID	UMPCA-UC-400
Use Case Name	Manage Announcement
Brief Description	This use case allow admin to create an announcement and they can edit or delete it.
Actor	Admin
Pre – Condition	Admin already login to the system.
Basic Flow	Admin

	1. The use case start when admin login to the UMP
	Confession Application.
	2. The application will display < <admin page="">&gt; interface.</admin>
	3. Admin click "Manage Announcement" button and
	application will redirect admin to < <manage< td=""></manage<>
	Announcement >> interface, all announcement data will be
	retrieve from the database and display at this page.
	4. Admin click "Make Announcement" button and
	application will redirect admin to < <make announcement<="" td=""></make>
	>> interface
	5. Admin input the content of announcement and click
	< <done>&gt; button and If other user login to the</done>
	application, this announcement will display on the < <user< td=""></user<>
	Page>>.
	6. Application will redirect back to "Manage
	Announcement".
Alternative Flow	Al. Fdit Announcement
Alternative Flow	A1. Eun Announcement
	1. Continue at step 3 of basic flow.
	2. Admin select the announcement that want to update and
	click "Edit" button to redirect to < <edit announcement="">&gt;</edit>
	interface.

	3. Next, Admin edit the content and click "Done" button to
	update the announcement to database.
	4. Application will redirect back to < <manage< td=""></manage<>
	Announcement >> Page.
	A2: Delete Announcement
	1. Continue at step 3 of basic flow.
	L L L L L L L L L L L L L L L L L L L
	2. Admin select the announcement that they want to delete
	and click "Delete" button.
	3 Data of announcement will be remove from the database
	5. Duti of announcement will be remove from the database.
	4. Application will redirect to << Manage Announcement >>.
Post-Condition	The announcement can be manage by admin successfully.

Table 3. 3. 4: Use case description of manage Announcement

## Manage order type

# Use Case Diagram



Figure 3. 3. 7: Use case for manage order type

Use Case ID	UMPCA-UC-500
Use Case Name	Manage order type
Brief Description	This use case allow admin to create new order type to let user choose and manage the order that created by other user.
Actor	Admin
Pre – Condition	Admin already login to the system.

Basic Flow	Admin
	1. The use case start when admin login to the UMP
	Confession Application.
	2. The application will display < <admin page="">&gt; interface.</admin>
	3. User click "Manage order type" button.
	4. The application will redirect user to << Manage order type >> interface.
	5. Admin click "Add new order type" button to add a new type of order for user to choose.
	6. Application will redirect to < <add new="" order="" type="">&gt; and admin need to input detail of new order type.</add>
	7. Admin click "Add" to save data to database and it will redirect to << Manage order type >> interface.
Alternative Flow	A1: Delete order type
	<ol> <li>Continue at step 4 of basic flow.</li> <li>Admin select the order type that they want to remove and click "Delete" button.</li> </ol>

	3. All data related to this order type will be remove from
	database and application will redirect to << Manage order
	type >> interface.
Post-Condition	Order type which will be available for user to make order by
	the user can be manage by admin successfully.

Table 3. 3. 5: Use case description of manage order type
### Manage order

# Use Case Diagram





### **Use Case Description**

Use Case ID	UMPCA-UC-600
Use Case Name	Manage order
Brief Description	This use case allow user to choose the order type such as order food from cafeteria or Rent for badminton court only if these order type is added by admin.
Actor	User
Pre – Condition	1. User already login to the system.

	2. Admin already add the type of order that the user want to			
	make order.			
Dasia Flam				
Basic Flow	User			
	1. The use case start when user login to the UMP			
	Confession Application.			
	2. The application will display < <user page="">&gt; interface.</user>			
	3. User click "My order" button and application will redirect			
	to << Manage order >> interface.			
	4. Data of ander that make her the year will be display of			
	4. Data of order that make by the user will be display at			
	< <manage order="">&gt;interface.</manage>			
	5. Next, user click "Add order" button and the application			
	will redirect to < <add order="">&gt; interface.</add>			
	6. User select the type of order that they want and input the			
	details of order.			
	7. User click "Done" button and the data of that order will			
	be save into database.			
	8 Application will redirect to << Manage order> interface			
	8. Application will reduced to << Manage order>> interface.			
Alternative Flow	A1: Delete order			
	1. Continue step 4 at basic flow.			
	· · · · · · · · · · · · · · · · · · ·			

2. User select the order that want to delete and details of that
order will be display.
3 User click "Delete" button to delete it and system will
redirect back to < </td
redirect back to << manage order>>interface.
A2: Manage order from user
1. The user which is the one who is responsible for the order
login to the UMP Confession Application.
2 The application will display < <user page="">&gt; interface</user>
2. The appreation will display < <es of="" tage="">&gt; interface.</es>
3. User click "My order" button.
4. The application will redirect user to << Manage order >>
interface and user click "Manage order" button.
5. The application will redirect user to $<<$ Insert order type
id >> interface
6. The user need to input the id of the order type that they
want to manage and click "Done" button then system will
redirect to << Manage order from user>> interface.
7. System will display all the order which is related to that
type of order.

	8. The user can choose order to view and input comment and click "Comment" button to update the progress of the order.			
	9. Next, it will redirect back to << Manage order from user >> interface.			
Post-Condition	An order will be available for add, view, delete and being updated in the database.			

Table 3. 3. 6: Use case description of manage order

# Manage Complaint

# Use Case Diagram



Figure 3. 3. 9 : Use case for manage complaint

### Use Case Description

Use Case ID	UMPCA-UC-700
Use Case Name	Manage Complaint
Brief Description	This use case allow user to make complaint and the admin can view it and reply on these complaint.
Actor	User and admin

Pre – Condition	User had register an account and already
	login to the system.
Basic Flow	Usor
Dasie 110w	User
	1. The use case start when user login to
	the UMP Confession Application.
	2. The application will display < <user< td=""></user<>
	Page>> interface.
	3 User click "My Complaint" button
	and system will redirect to My
	complaint> interface
	comprant// interface.
	4. A list of complaint that user made
	before will be retrieve from database and
	display at this page.
Alternative Flow	A1: Add Complaint
	1. Continue at step 4 in Basic flow.
	2. User click "Make Complaint" button.
	3. The application will redirect user to
	< <make complaint="">&gt; interface.</make>
	4. User input the content of complaint
	and click "Complaint Now" button to save
	data into database.

	A2: Delete Complaint (User)
	1. The use case continue at step 4 in basic flow.
	2. User select complaint that want to delete and click "Delete" button.
	A3: Manage complaint (Admin)
	1. After login as admin, admin can click "Manage Complaint" button in < <admin Page&gt;&gt; Interface and application will redirect admin to &lt;&lt; Manage Complaint&gt;&gt; interface.</admin 
	2. Admin select the complaint that they want to reply, and input reply on that complaint.
	3. Admin click "Reply" button and application will redirect to < <admin Page&gt;&gt;.</admin 
Post-Condition	The complaint can be manage by the user and admin successfully.

Table 3. 3. 7: Use case description for manage complaint

## 3.3.7 Activity diagram



Figure 3. 3. 10: Activity diagram of user



Figure 3. 3. 11: Activity diagram of admin

#### 3.4 Data Design

This section will describe all the data related to the proposed project include entity relationship diagram (ERD) and Database Dictionary.

#### 3.4.1 Entity relationship diagram



Figure 3. 4. 1: Entity relationship diagram of UMP Confession Application

# 3.4.2 Database Dictionary

Field Name	Description	Data Type	Constraint
ID	ID of user	INT	РК
username	Username	VARCHAR(20)	
password	Password for login	VARCHAR(20)	
HP	Phone number of the user	VARCHAR(12)	
Image	Image upload for admin to check to approve them as valid user	IMAGE	
validornot	To check wheatear the user is valid or not which 1 is valid and 0 is not valid	INT	

Table 3. 4. 1: Data Dictionary for User

Field Name	Description	Data Type	Constraint
ID	ID of Admin	INT	PK
username	Username	VARCHAR(20)	
password	Password for login	VARCHAR(20)	

Table 3. 4. 2: Data Dictionary for Admin

Field Name	Description	Data Type	Constraint
Post_ID	ID of post	INT	РК
TypeOfPost	Type of post that user make	VARCHAR(50)	
ContentOfPost	Content of post	VARCHAR(50)	
ID	ID of the user who make the post	INT	FK

Table 3.	4.	3:	Data	Dictionary	for	Post
1 4010 5.	•••	<i>·</i> ··	Data	Dictional	101	1 000

Field Name	Description	Data Type	Constraint
Comment_ID	ID of comment	INT	РК
ContentOfComment	Content of comment	VARCHAR(50)	
Post_ID	ID of post that have this comment	INT	FK
ID	ID of the user who make the comment	INT	

Table 3. 4. 4: Data Dictionary for Comment

Field Name	Description	Data Type	Constraint
Announcement _ID	ID of announcement	INT	РК
ContentOf Announce	Content of announcement	VARCHAR(50)	

Table 3. 4. 5: Data Dictionary for Announcement
---

Field Name	Description	Data Type	Constraint
Order_ID	ID of order	INT	РК
Order_Type_ID	Type of order that user choose	INT	FK
OrderType	Type of order	VARCHAR(50)	
DetailsOfOrder	Details of the order that user want to make	VARCHAR(50)	
Comment	Comment that used to update status of the order by the user who responsible to this order type.	VARCHAR(50)	
ID	ID of the user who make the order	INT	FK

Table 3.	4. 6:	Data	Dictionary	for	Order
----------	-------	------	------------	-----	-------

Field Name	Description	Data Type	Constraint
Order_Type_ID	ID of order type	INT	РК
OrderType	Type of order	VARCHAR(50)	

Field Name	Description	Data Type	Constraint
Complaint_ID	ID of complaint	INT	РК
ContentOfComplaint	Content of complaint that user want to write.	VARCHAR(50)	
Status	Status of complaint which is PENDING when user make the complaint and RECEIVE when admin reply the complaint.	VARCHAR(50)	
Reply	Reply from admin on a specific complaint.	VARCHAR(50)	

Table 3. 4. 8: Data Dictionary for Complaint

#### 3.5 **Proof of Initial Concept**

This section will describe all the data related to the proposed project which is design prototype of UMP Confession Application. A prototype is an early sample, model or release of a product created to test a concept or process. Figma is used as a tool to develop the prototype.

#### 3.5.1 Design Prototype



Figure 3. 5. 1: Login interface of UMP Confession Application



Figure 3. 5. 2: LoginAsAdmin interface of UMP Confession Application

Register		
Username		
Password		
Upload an Image to prove you are from UMP		
Phone number		
Re	gister	

Figure 3. 5. 3: Register interface of UMP Confession Application

USER	USER PAGE			
	User Main Page			
	My Post	My order		
	Make Post	My Complaint		
	Content of	announcement		
VIEW POST				

Figure 3. 5. 4: UserPage interface of UMP Confession Application



Figure 3. 5. 5: AdminPage interface of UMP Confession Application

Check account			
User who are still not invalid			
username	Image upload for admin to approve as valid user	Approve Reject	
username	Image upload for admin to approve as valid user	Approve Reject	

Figure 3. 5. 6: CheckAccount interface of UMP Confession Application

Manage Post		
LIST OF POST		
Content of post	Edit	
Content of post	Delete	
Content of post	Edit	
Content of post	Delete	
	_	

Figure 3. 5. 7: ManagePost interface of UMP Confession Application

Make Post		
Make a Post now !		
Type of post:		
Write anytthingd you want :		
Upload image for post :		
Ма	ke Post	

Figure 3. 5. 8: MakePost interface of UMP Confession Application



Figure 3. 5. 9: EditPost interface of UMP Confession Application



Figure 3. 5. 10: ViewPost interface of UMP Confession Application

Manage Order UMP Con	Ifession Ap	plication	
Add order	Filter by	▼	Done
L	IST OF ORE	DER	
Content of or	der	Delet	e
Content of order		Delete	

Figure 3. 5. 11: ManageOrder interface of UMP Confession Application



Figure 3. 5. 12 : AddOrder interface of UMP Confession Application



Figure 3. 5. 13: ViewOrder interface of UMP Confession Application

Manage Announcement
Make Announcement
List of Announcement
Content of Announcement
Edit Delete

Figure 3. 5. 14: ManageAnnouncement interface of UMP Confession Application



Figure 3. 5. 15: MakeAnnouncement interface of UMP Confession Application



Figure 3. 5. 16: EditAnnouncement interface of UMP Confession Application

Manage order type
Add new order type
Order type that available now
CAFE KK5 Delete
HALL RENT Delete

Figure 3. 5. 17: ManageOrderType interface of UMP Confession Application



Figure 3. 5. 18: AddOrderType interface of UMP Confession Application

Admin Manage Post
LIST OF POST
Content of post Delete
Content of post Delete

Figure 3. 5. 19: AdminManagePost interface of UMP Confession Application

UMP Confession Appl	ication
Make Complain	ıt
LIST OF Compla	int
Content of Complaint	Status
Reply from Admin Delete	

Figure 3. 5. 20: My Complaint interface of UMP Confession Application



Figure 3. 5. 21: MakeComplaint interface of UMP Confession Application

Manage Complaint				
Complaint received from user				
content of complaint				
Relpy from admin Update				
content of complaint				
Relpy from admin Update				

Figure 3. 5. 22: Manage Complaint interface of UMP Confession Application


Figure 3. 5. 23: add comment interface of UMP Confession Application

# 3.6 Testing plan

This section will describe the testing strategy or any approach that will be use on testing phase of UMP confession application. To prepare for testing phase that will be conduct, User Acceptance Test (UAT) form is prepared for future use.

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register			
2		Login as admin			
3		Login as user			
4		Check account			
5	Manage post	Make post			
6		Edit post			
7		Delete post			
8	Add comment	Manage comment			
9	Manage order	Make order			
10		Edit order			
11		Delete order			
12		View order and give			
		comment			
13	Manage announcement	Make announcement			
14		Edit announcement			
15		Delete announcement			
16	Manage order type	Add new order type			
17		Delete order type			
18	Manage Complaint	Make complaint			
		Reply complaint			

This test has been performed by:

Name:

Signature:

Date:

# UAT Form use for testing

## 3.7 Potential Use of Proposed Solution

This section will give explanation of potential use of proposed solution in real time situation. My proposed application is a social media application that only specific to Universiti Malaysia Pahang. The purpose of this application is to help people to build relationship with each other. Social media is about connecting people to people. It provides a platform for people to speak up what inside them. However, it not just only can use by Universiti Malaysia Pahang, other community such as university or company can use this application as a template to build one similar for same purpose. It has a big potential that this type of application can widely use by different community.

#### **CHAPTER 4**

#### **IMPLEMENTATION, RESULT AND DISCUSSION**

### 4.1 Introduction

This chapter discuss about the implementation, result or any findings of the project. This chapter contains the result of the finding based on the experiment or testing that has been done. Its also includes the explaination of the discussion that shows the objectives of the project is fullfilled.

In this chapter, it will discuss about the implemented interfaces and all functions in the UMP Confession Application. All the detailed and accurate information of the application will be display by figures and explaination as the solution to the problem stated in Chapter 1. Lastly, the testing strategy of UMP Confession Application is the User Acceptance Test (UAT) report which stated in the previous chapter.

## 4.2 Implementation Process

In this section, it will discuss in details on the implementation process that involve in completing UMP Confession Application from the beginning to the end of the development process. All of the user interface will be shown in this chapter and description for each user interface will be included. Besides, some part of the code for this application and database environment for this application also will be include in this section.



Figure 4. 1: Interface of Main Page of Application

The figure show the interface of main page which will display first when the user open the application with their phone. The user can choose to login, register or login as admin in this interface,

半夜12:38 🏵 🖯 🕄	\$ .ıll <sup>Vo</sup> 🧙 💷 ≁			
Register Page	e)			
UMP Confession Application				
×	Your Username			
	Your Password			
C	Your Phone Number			
Please upload UMP. You can o image	image that prove you are from lick 📷 to take photo of your			
Ō.	CLICK HERE TO UPLOAD 📁			
	REGISTER			
	•			

Figure 4. 2: Interface of register page

The figure show the interface of register page. This interface will allow the user to register an account so they can keep using the application. The user need to input their username, password, an image to prove that the user is from UMP (for example: their student card), and their phone number.

半夜12:52	半夜12:52 句 당 🛛 당 🕴 🗱 🙀 🖗 🕫				
😊 tes	😊 tester1 : User Main Page				
UMP Confession Application					
	MY POST		MY ORDER		
	MAKE POST		MY COMPLAINT		
This is confes: Please anythin To all I	This is first announcement from UMP confession Application. Please enjoy to use this application for post anything,Thanks.				
Thanks for download and using this app Please feel free to make any post.					

Figure 4. 3: Interface of user main page

The figure show the interface of the user main page which is the page that system will redirect to after the user is enable to login to the system. All function that the user can be select is include at this page. Besides, the announcements from UMP or admin also will be display at this page. The user can click "View Post" button to view all the posts that post by all user.



Figure 4. 4: Interface of make post page

The figure show the interface of the user when they want to make a post or confession. They need to select the type of their post, content of their post and upload an image if they want to include in the post. When they finish input all the data, they can click "DONE" button to post it. A post successful message will pop up to the user.



Figure 4. 5: Interface of my post page

The figure show the interface of user when they click "My post" button at their main page. This interface will only display all the post that they make and they can click on it to delete it or edit it.

凌	晨1:13 🞯 号 😝 号		\$
😅 tester1 : View Post			
UMP			
	List of	Post	
	Select type of post	•	SEARCH
se	em 7 is coming to enddd 😭		
1(	00% can go back 😍		
Fi	rst anime i watch in my life		
go	bod		
		)	•

Figure 4. 6: Interface of view post page

The figure show the interface of user when they click "View post" button at button of their main page. This interface will display all the post that all the users make and they can click on it to view in details or comment it. They can also use the dropdown menu and search button to view the post according to the type of post.



Figure 4. 7: Interface of view post by using filter

The figure show the interface of user when they use the dropdown menu and search button to view the post according to the type of post.



Figure 4. 8: Interface of my complaint

The figure show the interface of user when they click "My complaint" button at user main page. This interface will allow the user to view the complaint that they make before and they can always click on the complaint to view reply from the admin if the status change from PENDING to DONE.



Figure 4. 9: Interface of my complaint

The figure show the interface after the user click on the complaint to view reply from the admin if the status change from PENDING to DONE. The user can delete it anytime when they want to delete it.

凌晨1:12 🎯 🖯 🗗 🖯	💲 . 11    😽 🥱 🎟 🕫		
😊 tester1 : Maki	😊 tester1 : Making Complaint		
UMP Confes	ssion Application		
Any compla	aint about UMP?		
Content of Comp	laint:		
why cafe got so ma	any cat? ?		
СОМР	LAINT NOW		
	•		

Figure 4. 10: Interface of make complaint

The figure show the interface of user when they click "Make complaint" button at My complaint page. This interface will allow the user to write and send a complaint to the admin of the application to view.



Figure 4. 11: Interface of View post (On single post)

The figure show the interface of user when they click on the post that they want to view in View post page. They can always give comment to the post by input comment and click the "comment" button.



Figure 4. 12: Interface of View comments

The figure show the interface of user when they click "view comment" button in any post that they view. It will display a list of comments that given to the selected post.



Figure 4. 13: Interface of my order

The figure show the interface of user when they click on "My order" button at the user main page. This page allow the user to manage their order and a list of order that the user made before will be display at this page.

半夜12:56 🎯 🖯 🖪 🖯	∦ .11   ₩FFi 🛜 🎯 +			
😊 tester1 : Making or	😅 tester1 : Making order			
UMP Confession	Application			
What do you want to	o make order? 📋			
CAFE JIMMY				
PS5 game station				
Library Kbox room				
BADMINTON COURT				



Figure 4. 14: Interface of make order

The figure show the interface of user when they click "make order" button at my order page. It will display a list of option for user to select by the user for making order.

半夜12:57 🎯 🖯	9	🖇 🚛 🖓 🙀 🖓 🙀 🖗	
😊 tester1 : N	😊 tester1 : Making order		
UMP Confession Application			
What do yo	u want to mak	e order? 📋	
	CAFE JIMMY		
Details of order:			
One tea ice, kahchun	take at 615pm		
	DONE 🔽		
	۲	•	

Figure 4. 15: Interface of make order

The figure show the interface of user when they already choose which order type that they want to make. They can input any details that related to their order and click "DONE" button to make order.



Figure 4. 16: Interface of view order

The figure show the interface of user when they want to view the details and comment from the person that in charge of the order that they make. The user can also delete the order anytime they want to delete or cancel it.



Figure 4. 17: Interface of insert order type ID

The figure show the interface of user when they want to manage the order of their owned cafeteria or any facilities after they click "Manage order" at my order page. They need to input the password of their order type which is the order type ID so they can have access to view the order from other user and manage it.



Figure 4. 18: Interface of manage order

The figure show the interface of user when they insert a valid and correct order type ID to manage the order of their owned cafeteria or any facilities. A list of order which is belong to that order type will be display at this page so the user can manage them.

凌晨1:10 🎯 🖯 <table-cell></table-cell>	5 <b>∦ .</b> ıll <sup>Vo</sup> <b>? </b> 95 <i>+</i>	
😊 tester1 : Manage order		
UMP Confession Application		
Order from:	tester1	
Order type:	CAFE JIMMY	
Order Details:	One tea ice, take at 615pm kahchun	
Comment:	noted	
noted		
	REPLY	
	•	

Figure 4. 19: Interface of manage order

The figure show the interface of user when they choose an order from the list of order. The user can always reply to the order so they can give update to the user who make this order so they can always know the latest status of their order.



Figure 4. 20: Interface of login page as admin

The figure show the interface of admin when they choose to login as admin at main page of application. The admin need to input the valid and correct username and password to access the system as an admin.



Figure 4. 21: Interface of admin main page

The figure show the interface of admin main page which will display first when the admin log in successfully to the system. They can choose to do different task and function as an admin.



Figure 4. 22: Interface of manage announcement

The figure show the interface when the admin want to manage the announcement after click "Manage announcement" button at admin main page. They can view the list of announcement that they make before and click on it to delete it if the announcement is already useless or expiry.



Figure 4. 23: Interface of make announcement

The figure show the interface when the admin want to make a new announcement after click "Make announcement" button at manage announcement page. They can input the content of the announcement and click done button to post it to all users.



Figure 4. 24: Interface of edit announcement

The figure show the interface when the admin want to edit one of the announcement after select from the list. They can edit the content of the announcement and click done button to update it to all users.

凌晨1:40 🞯 🖯	\$.⊪I\ <sup>v</sup> ‱ 🧙 🎟 ≁	
Admin: Manage Post		
UMP		
UMP Confession Application		
List of Po	ost	
First spipe i wetch in py life		
First anime i watch in my life		
100% can go back 😍		
100% can go back 😍		
good sem 7 is coming to enddd		



Figure 4. 25: Interface of admin manage post

The figure show the interface when the admin want to manage all the post that make by user of this application after they click "Manage post" button at admin main page. Admin can view them and delete it if they found out any post that are having problem.



Figure 4. 26: Interface of check account page

The figure show the interface when the admin want to manage the user account who are still invalid to use the application after click "Check account" button at admin main page The admin can click on to the user on the list to view their uploaded image for validation.



Figure 4. 27: Interface of check account (Approve account)

The figure show the interface when the admin want to approve the user account if the user are valid to use the application. When the admin click on to the user on the list at previous page, they can view their uploaded image for validation and click "approve" button to allow the user who registered can login to the system.

半夜12:49 🗑 🖯	0 9	\$	
Admin: Man	Admin: Manage Order Type		
UMP Confession Application			
ADD NEW ORDER TYPE			
L	ist of Order T	уре	
CAFE JIMMY			
PS5 game statio	on		
Library Kbox room			
BADMINTON CO	DURT		
BADMINTON CO	JURT		



Figure 4. 28: Interface of manage order type

The figure show the interface when the admin want to manage order type that will be available for user to choose for make order such as make order at café or booking for a badminton court in UMP after the admin click "Manage order type" button at admin main page.

半夜12:49 🎯 🖯 😚 🖯	\$.₁    <sup>Vo</sup> 奈 @ ≁	
Admin: Add New Order Type		
UMP Confession Application		
Add new ord	ler type	
Order type		
Order type ID		
ADI		



Figure 4. 29: Interface of add new order type

The figure show the interface when the admin want to add a new order type that will be available for user to choose for make order such as make order at café or booking for a badminton court in UMP. The user who want to manage the order need to find admin of this application so the admin can help them to add a new order type with an order type ID which will be the password for user to manage their order later.



Figure 4. 30: Interface of admin manage complaint

The figure show the interface when the admin want to manage the complaint given by the user after they click "Manage complaint" button at admin main page. All the complaint which still pending will display in this page and the admin can click on the complaint to view the complaint in a single page.



Figure 4. 31: Interface of admin manage complaint (Give reply to user)

The figure show the interface when the admin want to reply to the complaint that given by the user. After the admin can click on the complaint to view the complaint in a single page, they can also give their reply to the user and the status of the complaint will change from "pending" to "done".

## Code

Code below is part of the function for user to make post with their image.

```
AdapterView.OnItemClickListener() {
parent.getItemAtPosition(position).toString();
            reference.addListenerForSingleValueEvent(new
ValueEventListener() {
                public void onDataChange(@NonNull DataSnapshot
```
```
public void onFailure(@NonNull Exception e) {
Toast.LENGTH LONG).show();
                public void onCancelled(@NonNull DatabaseError error)
```

Code below is part of the function for user to view all post that the other user post to this application.

```
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   getSupportActionBar().setTitle("\uD83D\uDE0A "+name+" : View
   btn c.setVisibility(View.INVISIBLE);
   ump.setVisibility(View.VISIBLE);
   date = findViewById(R.id.post id);
   userImage = findViewById(R.id.image);
```

```
parent.getItemAtPosition(position).toString();
listview.getItemAtPosition(position);
            String imageValid = orderClass.getStorageReference();
FirebaseStorage.getInstance("gs://ump-
                    public void onSuccess(Uri uri) {
```

```
databaseReference.addValueEventListener(new
ValueEventListener() {
                public void onDataChange(DataSnapshot snapshot) {
                public void onCancelled(DatabaseError firebaseError) {
```

```
FirebaseDatabase.getInstance("https://ump-confession-default-
rtdb.asia-southeast1.firebasedatabase.app/");
ValueEventListener() {
                          public void onDataChange(@NonNull DataSnapshot
                               commentclass.setId(CommentID);
String.valueOf(TimeUnit.MILLISECONDS.toSeconds(System.currentTimeMilli
                                        Toast.makeText(ViewPost.this,
                                        startActivity(intents);
                               }).addOnFailureListener(new
OnFailureListener() {
                                   public void onFailure(@NonNull
Exception e) {
                                       Toast.makeText(ViewPost.this,
                          public void onCancelled(@NonNull DatabaseError
```

```
protected void onStart() {
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                Collections.reverse(userList);
            listview.setAdapter(postAdapter);
Toast.LENGTH SHORT).show();
```

In the process of develop this UMP Confession Application, Android Studio which is an integrated development environment used to develop android application is being used. Java language is used as the programming language in Android Studio to build the proposed mobile application. Besides that, a cloud service which is Firebase will be used together with Android Studio. The Firebase real time database is a cloud-hosted database. All data that will be store and retrieve will be stored as JSON and synchronized in real time to every connected user. All of user who using this application will share one real time database instance and automatically receive updates with the newest data.

🞽 Eile Edit View Navigate Code Refact		
UMPCA $\rangle$ app $\rangle$ src $\rangle$ main $\rangle$ java $\rangle$ com $\rangle$ ex		e di 🖂 🛎 😘 🐢 👸 🗏 🟘 🖬 🗛 🛛 🗛 📄 🦷
💆 🛎 Android 👻 😌 🔄 🛨 🗢 —	🛛 ew_Post_in_type.java 👋 🌀 EditAnnouncement.java 👋 🏭 activity_edit_announcement.xml 👋 🥥 MakeAnnouncem	nt.xml × 🕒 MakePost.java × 💿 ViewPost.java × 🛔 🗸 💉
AdOnderType     AdonindargeFost	<pre>package com.example.umpca;  package com.example.umpca;  public class ViewPost extends AppCompatActivity {     public class ViewPost extends AppCompatActivity {         private String name,names,texts;         private String name,names,texts;         private String name,names,texts;     } } </pre>	
AdminPage     Back     CheckAccount     CommentAdapter     CommentClass     commentClass	and     private istrang connectations, connectation in dances,       51     private istrang connectations;       52     private istrangenetClass;       53     StorageReference storageReference;	
<ul> <li>Complant/Usagter</li> <li>Complant/Las</li> <li>EditAnouncement</li> <li>EditPost</li> <li>Insert, order_type_id</li> <li>LoginAsAdmin</li> <li>MainActivity</li> </ul>	55     private DatabaseReference reference, reference2;       56     FirebaseDatabase firebaseDatabase2;       57     Query re;       58     private postAdapter postAdapter;       59     private commentAdapter commentAdapter;       60     60	
MiskeAnnouncement     MakeAnnouncement     MakeComplaint     MakeComplaint     ManageComplaint     ManageComplaint     ManageCorderformuser     ManageCorderformuser     ManageCorderformuser     ManageCorderformuser	61     ListView Listview, 2:       62     ListView Listview, 2:       63     TextView date, date2, textview;       64     TextView comment_content;       65     LangeView userlange, uno;       66     CommentClass commentLass;       67     TextInputLayout textInputLayout;       68     Listview date date date	= - -
MyComplaint     NoticeAdapter     OnticeClass     Opening     OrderAdapter     OrderAdapter	79 Constraintlayout blogger_layout,blogger_layout2; 71 Button btn_accept,btn_c,btn_search; 73	Converting Barrier
Failed to start monitoring b6afcf8eb399 (today	y 9.05 PM)	70:52 LF UTF-8 4 spaces 🎦 🛄

Figure 4. 32: Android Studio

👌 Firebase	UMP Confession 🔻	转到文档	٤	9
♠ 项目概览	实时数据库			?
项目快捷方式	数据 规则 备份 使用量			
Realtime Database     Authentication     Firestore Database	GD https://ump-confession-default-rtdb.asia-southeast1.firebasedatabase.app	۵ :	ž	:
In Storage ② App Check 产品类别	https://ump-confession-default-rtdb.asia-southeast1.firebasedatabase.app/ - announcements - comments			
构建 🖌	<ul> <li>complaints</li> <li>order</li> </ul>			
发布与监控 <b>&gt;</b> 分析 <b>&gt;</b>	Ordertype Optimized and the second			
吸引 🗸	() – users			
<b>##</b> 所有产品				
Spark 升級 免费 \$0/月	◆ 教授希心里:新加坡(asis-southeast1)			•

Figure 4. 33: Real-time database for the application

#### 4.3 Testing and Result Discussion

In this section, the main purpose is to represent and design the process of UMP confession application's User Acceptance Test (UAT). According to previous chapter and the testing strategy. User Acceptance Test (UAT) form will prepare for future use and use on testing phase of UMP confession application.

Each module is designed and implemented based on the interfaces of application, and all the functions, text view, input form, buttons, data displayed and others had been tested as much as possible for each interface. The first test cases and second test cases are shown in the table below. The system have been test multiple times after all of the module is implemented successfully.

Besides, 5 student who study in Universiti Malaysia Pahang which is also the user scope for this application are found and they had download it to test on using the application. During the testing process, they also had complete the UAT form that being prepared before. All these UAT form are included in appendix C.

#### **CHAPTER 5**

#### CONCLUSION

## 5.1 Introduction

In this chapter, it will conclude the project that have been done, which consists of conclusion of the project. This thesis consists of five chapters. Chapter 1 discussed the project idea, objective, problem statement scope of the proposed application. Chapter 2 discussed the other existing application which related to proposed application. Comparisons are made between these three existing application and the proposed application in this chapter. Chapter 3 discussed the methodology, the application structure and architecture with diagrams. The design of user interface shown in this chapter. In Chapter 4, it discussed the outputs and results after the proposed application is developed. In this Chapter 5, the limitations of the application and future work can be done to improve the application will be discussed.

#### 5.2 **Objective Revisited**

As stated in Chapter 1, there are three objectives of this project. There are to study the existing application related to speak up what inside the user by posting message on a platform , to develop a social media application using Android studio and become a platform for the student in UMP to speak out their emotion and chat with others that may have same interest so they can reduce their stress and to evaluate the functionality of UMP Confession application. The first objective is achieved where the existing applications are chosen and discussion with pros and cons. The second objective also achieved where the UMP Confession application is built at the end of the project. The last objective is achieved as well where the functionalities of UMP Confession application are evaluated with UAT form.

### 5.3 Limitation

1. Language Constraint

The programming language that has been using in this project is by using JAVA. JAVA is one of the language that used for developing an Android mobile application in android studio. However, unlike the web application, which can be used on any platform. UMP Confession Application is not able to be run in IOS platform due to the constraint of programming language.

2. Data Constraint

User can make post in UMP Confession Application by upload an image to write post with image. However, this application is not able to be upload other format of data such as video and document due to constraint of data.

#### 5.4 Future Work

In this section, it will discuss suggestion and enhancement of UMP Confession Application, including knowledge and contribution to the university, faculty, society or writer throughout the project completion.

- I. Validation of account can be make improvement by implement AI which will approve account automatically after user scan their student id card or any document which can prove them are from UMP.
- II. This application can be widely used by other community, university or company so they can also have their own social media application.
- III. Develop this application in Swift language so that the IOS phone user can also use the application also.

#### REFERENCES

- UITM students die due to stress from studying: What is the Mohe doing to address this issue. Afterschool.my Higher Education Advisor in Malaysia. (n.d.). Retrieved June 5, 2022, from https://afterschool.my/articles/uitm-students-die-due-to-stress-from-studying-what-is-the-mohe-doing-to-address-this-issue
- *Ump confessions*. Facebook. (n.d.). Retrieved from https://www.facebook.com/UMPConfessions
- Mrchawla. (2015, March 24). 5 apps that let you express what you feel, anonymously #section66a. The Unbiased Blog. Retrieved from https://www.theunbiasedblog.com/2015/03/5-apps-that-let-you-express-whatyou-feel-anonymously-section66a.html
- X., W. V. C. (2016). Handbook of Research on Learning Outcomes and opportunities in the Digital age. Information Science Reference, an imprint of IGI Global.
- Google. (n.d.). Anonymous Social Secret Confessions & Chats Apps on Google Play. Google. Retrieved from https://play.google.com/store/apps/details?id=in.telling.ly&hl=en&gl=US
- Whisper. (n.d.). Retrieved from http://whisper.sh/
- Google. (n.d.). Speakup / Consumer Complaints apps on Google Play. Google. Retrieved from https://play.google.com/store/apps/details?id=com.dhwaniris.speakup
- Agile Software Development Lifecycle Phases explained. Relevant Software. (2022, May 20). Retrieved from https://relevant.software/blog/agile-softwaredevelopment-lifecycle-phases-explained/
- Ng, V. (2019, March 10). *Non-functional requirement of the mobile development system*. Medium. Retrieved from https://medium.com/@vishwasng/non-functional-requirement-of-the-mobile-development-system-e0ed98f2a872

# APPENDIX A SRS



# SOFTWARE REQUIREMENT **SPECIFICATION (SRS)**

[UMP Confession Application]

# **DOCUMENT APPROVAL**

	Name	Date
Authenticated by:		
Name		
Approved by:		
Client		

Software	:	

Archiving Place :

# TABLE OF CONTENT

# CONTENTPAGEDOCUMENT APPROVALiiTABLE OF CONTENT.iiiLIST OF FIGURESivLIST OF TABLESviiLIST OF APPENDICESvii1.1 PROJECT DESCRIPTION.11.2 SYSTEM IDENTIFICATION11.3 CONTEXT DIAGRAM21.4 DATA FLOW DIAGRAM32.1 USE CASE DIAGRAM AND DESCRIPTION42.2 SEQUENCE DIAGRAM243.1 INTERFACE DESIGN453.2 HARDWARE AND SOFTWARE SPECIFICATION68

# LIST OF FIGURES

Eigune 1. 2 Data flow diagram of UMD Confession Application
Figure 1. 2 Data now diagram of UMP Confession Application
Figure 2. 1: Interaction between UMP Confession application and the actors
Figure 2. 2: Use case for manage registration
Figure 2. 3 Use case for manage post
Figure 2. 4 Use case for manage comment
Figure 2. 5 Use case for manage Announcement
Figure 2. 6: Use case for manage order type
Figure 2. 7 : Use case for make order18
Figure 2. 8: Use case for make complaint21
Figure 2. 9: Manage registration Basic Flow Sequence Diagram
Figure 2. 10 Manage registration [Alternative Flow A1: Having account] Sequence Diagram
Figure 2. 11: Manage registration [Alternative Flow A2: Login as admin] Sequence Diagram
Figure 2. 12: Manage registration
Figure 2. 13: Manage registration [Alternative Flow A4: Check account (Approve account)] Sequence Diagram
Figure 2. 14: Manage registration [Alternative Flow A5: Check account (Reject account)] Sequence Diagram
Figure 2. 15: Manage registration [Exception Flow E1: wrong username and password or invalid account] Sequence Diagram
Figure 2. 16: Manage Post Basic Flow Sequence Diagram
Figure 2. 17: Manage Post [Alternative Flow A1: Create Post] Sequence Diagram
Figure 2. 18: Manage Post [Alternative Flow A2: Edit Post] Sequence Diagram
Figure 2. 19: Manage Post [Alternative Flow A3: Delete Post] Sequence Diagram

Figure 2.	20: Manage Post [Alternative Flow A4: Delete Post as admin] Sequence Diagram
Figure 2. Diagram	21: Manage Post [Alternative Flow A5: View in different type of post] Sequence
Figure 2.	22 : Manage comment Basic Flow Sequence Diagram
Figure 2.	23 : Manage announcement Basic Flow Sequence Diagram
Figure 2. Diagram	25: Manage announcement [Alternative Flow A1: Edit announcement] Sequence
Figure 2. Diagram	26: Manage announcement [Alternative Flow A2: Delete announcement] Sequence
Figure 2.	27: Manage order type Basic Flow Sequence Diagram
Figure 2.	28: Manage order type [Alternative Flow A1: Delete order type] Sequence Diagram
Figure 2.	29: Manage order Basic Flow Sequence Diagram40
Figure 2.	30: Manage order [Alternative Flow A1: Delete order] Sequence Diagram41
Figure 2.	31: Manage Order [Alternative Flow A2: View order] Sequence Diagram42
Figure 2.	32 : Manage Complaint Basic Flow Sequence Diagram43
Figure 2.	33: Manage Complaint [Alternative Flow A1: Delete complaint] Sequence Diagram 
Figure 2. Diagram	34: Manage Complaint [Alternative Flow A2: Reply Complaint] Sequence 44

Figure 3. 1 : Login interface of UMP Confession Application	.45
Figure 3. 2 : LoginAsAdmin interface of UMP Confession Application	.46
Figure 3. 3: Register interface of UMP Confession Application	.47
Figure 3. 4: UserPage interface of UMP Confession Application	.48
Figure 3. 5: AdminPage interface of UMP Confession Application	.49
Figure 3. 6: CheckAccount interface of UMP Confession Application	.50

# SOFTWARE REQUIREMENT SPECIFICATION (SRS) FKOM

Figure 3. 7: ManagePost interface of UMP Confession Application	51
Figure 3. 8: MakePost interface of UMP Confession Application	52
Figure 3. 9: EditPost interface of UMP Confession Application	53
Figure 3. 10: ViewPost interface of UMP Confession Application	54
Figure 3. 11 : ManageOrder interface of UMP Confession Application	55
Figure 3. 12: AddOrder interface of UMP Confession Application	56
Figure 3. 13: ViewOrder interface of UMP Confession Application	57
Figure 3. 14: ManageAnnouncement interface of UMP Confession Application	58
Figure 3. 15: MakeAnnouncement interface of UMP Confession Application	59
Figure 3. 16: EditAnnouncement interface of UMP Confession Application	60
Figure 3. 17: ManageOrderType interface of UMP Confession Application	61
Figure 3. 18: AddOrderType interface of UMP Confession Application	62
Figure 3. 19: AdminManagePost interface of UMP Confession Application	63
Figure 3. 20: My Complaint interface of UMP Confession Application	64
Figure 3. 21: Make Complaint interface of UMP Confession Application	65
Figure 3. 22: Manage Complaint interface of UMP Confession Application	66
Figure 3. 23: Add comment interface of UMP Confession Application	67

# LIST OF TABLES

Table 2. 1 : Use case description of manage registration	8
Table 2. 2 : Use case description for manage post	11
Table 2. 3 : Use case description for add comment	13
Table 2. 4: Use case description for manage Announcement	15
Table 2. 5: Use case description for manage order type	17
Table 2. 6: Use case description for make order	21
Table 2. 7: Use case description for manage complaint	23

Table 3.1:	Hardware Specification	
Table 3. 2:	Software Specification	69

# LIST OF APPENDICES

#### **CHAPTER 1**

#### **1.1 PROJECT DESCRIPTION**

UMP Confession application is an application that is develop for the social use of the people in a University and this UMP Confession application is aim for use by the student or anyone who stay in Universiti Malaysia Pahang. It will consist of 6 modules which are manage registration, manage post, add comment, manage announcement, manage order type and manage order.

In the manage registration module, user able to register for a new account so they can log into the application. Since the user need to register with an evidence that can prove that they are stay in UMP, admin is able to view all user details and remove user from the application or set user to be valid for using the application.

In manage post, user can make post by select the type of post that they want to post such as confession or complaint. Besides, they also manage to edit and delete their own post. Admin also have permission to delete any post if they want. The post that make will not showing the username. In add comment, the user can give comment to any post if they want. In manage announcement, the admin can make a announcement to let everyone view the announcement since the announcement will be on the top of the main page of user. Beside, admin is free to edit them or remove them from database.

In manage order and manage order type, it will be more complicated and these two module is related to each other. This application only provide a platform for the user who wish to make order and other user who need to receive the order. For example, the owner of badminton court can find and ask the admin to add a new order type so the order type of rent badminton court will be available for the user to choose. Then, other user can make order to rent for badminton court. Besides, the user who own the badminton court can view order from the application and leave comment to reply to the user who make order.

#### **1.2 SYSTEM IDENTIFICATION**

System Identification Number: UMPCA \_2022\_V1

System Name: UMP Confession Application

System Abbreviation: UMPCA

Year: 2022

Version: V1

#### **1.3 CONTEXT DIAGRAM**



Figure 1. 1 Context diagram of UMP Confession Application

Figure 1.1 shows the context diagram of the UMP Confession Application. Context diagram refers to DFD (data flow diagram) that makes details and boundaries of the designed system clearer that it makes project's scope will be simpler and straightforward. The context diagram consists of two entities who are user and admin who are going to interact with the application.

First, the user will need to insert user information and password to application so they can have completed register. The user who give correct username and password can have valid login and they can input any content of post or comment to the application and they also can view other post or comment from the application which will retrieve data. The user is also allowed to input order details into application and view order details from application. Beside, user also can view announcement that create by admin. The user also can make complaint to admin if they want.

For admin, they can login by input correct username and password into the application. After login successfully, admin can view all post, user information, announcement and order type from the application. Besides, admin can input new announcement and new order type data into the application. Admin is able to reply to user on their given complaint.



# 1.4 DATA FLOW DIAGRAM

Figure 1. 2 Data flow diagram of UMP Confession Application

#### **CHAPTER 2**

#### 2.1 USE CASE DIAGRAM AND DESCRIPTION



Figure 2. 1: Interaction between UMP Confession application and the actors

# 2.1.1 Manage Registration

# Use Case Diagram



Figure 2. 2: Use case for manage registration

Use	Case	Description
-----	------	-------------

Use Case ID	UMPCA-UC-100
Use Case Name	Manage registration
Brief Description	This use case allow user to register an account and login to use this application. The admin also can approve the user account so they can login into application or reject them.
Actor	User and admin
Pre – Condition	User did not log into their account and don't have any account yet.
Basic Flow	User 1. The use case start when user open the UMP Confession Application.

	2. The application will display << I or in>>
	2. The application will display << Login>>
	interface.
	3. User click "Register" button and application
	will redirect user to < <register>&gt; interface</register>
	4. User input all the required details of user.
	5. User click "Register" button.
	6. The application will display < <login>&gt;</login>
	interface.
Alternative Flow	
	A1: Having account (User)
	1 The second
	1. User of admin input their username and
	password and click "Login" button.
	2. The application will redirect user to < <user< th=""></user<>
	Page>> interface if username and password is
	valid and they are approve by the admin to
	register account
	A2: Login as admin (Admin)
	1. Admin click "Login as admin" button to
	redirect to login as admin interface.
	2. Admin input username and password of
	admin which is "admin" and "admin123".
	3. Admin click "Login" button.
	4. The application will redirect user to << Admin
	Page>> interface if username and password is
	valid.
	A3: Check account (View account) (Admin)
	1. Continue step 2 in A1.

	2. Admin click "Check account" button and
	redirect to << <check account="">&gt; interface.</check>
	3. User account who are not valid will be
	display.
	A 4. Check account (Approve account)
	A4: Check account (Approve account)
	(Admin)
	1 Continue stop 2 in A2
	1. Continue step 5 in A2.
	2. Admin can click on each user to check their
	unloaded image to see either they are valid to
	uploaded image to see entier mey are valid to
	register account or not.
	3 Click "Approve" button to set the user
	account value to use.
	4. Application redirect to < <check account="">&gt;</check>
	interface
	interface.
	A5: Check account (Reject account) (Admin)
	1. Continue step 3 in A2.
	2 Admin can click on each user to check their
	unloaded image to see either they are valid to
	uploaded image to see either they are valid to
	register account or not.
	3 Click "Reject" button to set the user account
	invalid to use and delete it from database.
Exception Flow	E1 : Wrong username and password or
P	invalid account
	1. User will redirect to < <login>&gt; nage.</login>
	1

	2. Use case continue to step 1 in Alternative
	flow (A1).
Post-Condition	User can register an account which valid to use
	and login to the application.

# Table 2. 1 : Use case description of manage registration

# 2.1.2 Manage Post

# Use Case Diagram





Use Case ID	UMPCA-UC-200
Use Case Name	Manage Post
Brief Description	This use case allow user to create a post for confession or
	complaint and they can edit or delete it. It also allow
	admin to delete post.
Actor	User and admin
Pre – Condition	User had register an account and already login to the
	system.
Basic Flow	User
	1. The use case start when user login to the UMP
	Confession Application.
	2. The application will display < <user page="">&gt; interface.</user>

	3. User click "View Post" button and system redirect to
	< <view post="">&gt; interface.</view>
	3. The system will display the post from the database at
	< <view post="">&gt; interface.</view>
Alternative Flow	A1: Create post (User)
	1. User continue at step 2 of basic flow and click "Make
	Post" button.
	2. The application will redirect user to < <make post="">&gt;</make>
	interface.
	3. User input the content of post , select type of post that
	the user want to choose, and upload image if they want to
	post together
	4. User click < <done>&gt; button to save data to database</done>
	and application will redirect to "User Page"
	A2: Edit post (User)
	1. User continue at step 2 of basic flow and click "My
	Post" button.
	2. The application will redirect user to < <manage post="">&gt;</manage>
	interface.
	3. User select the post that they want to edit and click
	"Edit" button to go to < <edit post="">&gt; Interface.</edit>
	4. User edit the content and click "Done" button to update
	the post.
	5. Application will redirect to <>
	interface.
	A3: Delete post (User)

	1. User continue at step 2 of basic flow and click "My
	The set continue at step 2 of basic now and check ivity
	Post" button.
	2. The application will redirect user to < <manage post="">&gt;</manage>
	interface.
	3. User select the post that they want to delete and click
	"Delete" button.
	4. Application will redirect to << Manage Post>>.
	A4: Delete post as admin (Admin)
	1. After login as admin, admin can click "Manage Post"
	button in < <admin page="">&gt; Interface and application will</admin>
	redirect admin to << Admin Manage post>> interface.
	2. Admin select the post that they want to delete and click
	delete button.
	3. Application will redirect to < <admin page="">&gt;.</admin>
	A5: View different type of post (User)
	1. User continue at step 3 of basic flow, the user can click
	dropdown list menu to select type of post that they want
	to view.
	2. Click < <search>&gt; button after select the type of post</search>
	that user want and system will redirect to < <view in<="" post="" td=""></view>
	type>> interface.
	3. System will display all post that the type of post is
	same as user option.
Post-Condition	I ne post can be manage by the user and admin
	successfully.

# Table 2. 2 : Use case description for manage post

#### 2.1.3 Manage Comment

Use Case Diagram



Figure 2. 4 Use case for manage comment

Use Case ID	UMPCA-UC-300
Use Case Name	Manage Comment
Brief Description	This use case allow user to manage comment on any post.
Actor	User
Pre – Condition	User had register an account and already login to the system.
Basic Flow	User

	1. The use case start when user login to the
	UMP Confession Application and system will
	redirect to < <user page="">&gt; interface.</user>
	2. User click "View Post" button and system
	will redirect to < <view post="">&gt; interface.</view>
	3. The system will display the post from the
	database.
	4. The user choose the post that want to make
	comment and will redirect to < <view post="">&gt;</view>
	interface.
	5. The user insert the content of comment and
	click "Comment" button.
	6. The application will redirect to << View
	Post>> interface.
Alternative Flow	A1 View comment
	1. Continue Step 3 in basic flow.
	2. Click on "View comment" button and
	system will redirect to << view comment >>
	interface.
	3. All comment of the selected post will be
	retrieve from database and display at <<
	view comment >> interface.
Post-Condition	The comment will be added below the post.



#### 2.1.4 Manage Announcement

# Use Case Diagram





Use Case ID	UMPCA-UC-400
Use Case Name	Manage Announcement
Brief Description	This use case allow admin to create an announcement and they can edit or delete it.
Actor	Admin
Pre – Condition	Admin already login to the system.
Basic Flow	Admin
	1. The use case start when admin login to the UMP
	Confession Application.
	2. The application will display < <admin page="">&gt; interface.</admin>

	3. Admin click "Make Announcement" button.
	4. The application will redirect user to < <make< td=""></make<>
	Announcement >> interface
	5. Admin input the content of announcement and click
	< <done>&gt;&gt; button and If other user login to the application,</done>
	this announcement will display on the < <user page="">&gt;.</user>
	6 Application will redirect back to "Admin Page"
	o. Approation win reduced block to Akanin Fuge
Alternative Flow	A1: Edit Announcement
	1. Continue at step 3 of basic flow.
	2. Admin select the announcement that want to update and
	click "Edit" button to redirect to < <edit announcement="">&gt;</edit>
	interface.
	3. Next, Admin edit the content and click "Done" button to
	update the announcement to database.
	4. Application will redirect back to < <manage< td=""></manage<>
	Announcement >> Page.
	A2: Delete Announcement
	1. Continue at step 3 of basic flow.
	2. Admin select the announcement that they want to delete
	and click "Delete" button.
	2 Data of announcement will be some one from the database
	5. Data of announcement will be remove from the database.
	4. Application will redirect to << Manage
	Announcement >>.
Post Condition	The encoursement can be manage by the user and educin
	successfully
	successfully.

 Table 2. 4: Use case description for manage Announcement

# 2.1.5 Manage order type

#### Use Case Diagram





Use Case ID	UMPCA-UC-500
Use Case Name	Manage order type
Brief Description	This use case allow admin to create new order type to let user choose and manage the order that created by other user.
Actor	Admin
Pre – Condition	Admin already login to the system.
Basic Flow	Admin <ol> <li>The use case start when admin login to the UMP Confession Application.</li> <li>The application will display &lt;<admin page="">&gt; interface.</admin></li> <li>User click "Manage order type" button.</li> </ol>
4. The application will redirect user to << Manage order	
---	
type >> interface.	
5. Admin click "Add new order type" button to add a	
new type of order for user to choose.	
6. Application will redirect to < <add new="" order="" type="">&gt;</add>	
and admin need to input detail of new order type.	
7. Admin click "Add" and it will redirect to < <admin< td=""></admin<>	
Page>> interface.	
A1: Delete order type	
1. Continue at step 4 of basic flow.	
2. Admin select the order type that they want to remove	
and click "Delete" button.	
3. All data related to this order type will be remove	
from database and application will redirect to <<	
Manage order type >> interface	
Wanage order type >> interface.	
Order type which will be available for user to make	
order by the user can be manage by admin successfully.	
,	

 Table 2. 5: Use case description for manage order type

# 2.1.6 Manage order

# Use Case Diagram



Figure 2. 7 : Use case for make order

# **Use Case Description**

Use Case ID	UMPCA-UC-600
Use Case Name	Manage order
Brief Description	This use case allow user to choose the order type such as order food from cafeteria or Rent for badminton court only if these order type is added by admin.
Actor	User
Pre – Condition	<ol> <li>User already login to the system.</li> <li>Admin already add the type of order that the user want to make order.</li> </ol>
Basic Flow	User

1. The use case start when user login to the
UMP Confession Application.
2. The application will display < <user page="">&gt; interface.</user>
3. User click "My order" button and application will redirect to << Manage order >> interface.
4. Data of order that make by the user will be display at < <manage order="">&gt;interface.</manage>
5. Next, user click "Add order" button and the application will redirect to < <add order="">&gt; interface.</add>
6. User select the type of order that they want and input the details of order.
7. User click "Done" button and the data of that order will be save into database.
8. Application will redirect to << Manage order>> interface.
A1: Delete order
1. Continue step 4 at basic flow.
2. User select the order that want to delete and details of that order will be display.
3. User click "Delete" button to delete it and system will redirect back to < <manage order&gt;&gt;interface.</manage 

A2: Manage order from user
1. The user which is the one who is
1. The user which is the one who is
responsible for the order login to the UMP
Confession Application.
2. The application will display < <user< th=""></user<>
Page>> interface.
3. User click "My order" button.
-
4 The application will redirect user to <<
4. The appreation will reduced user to <<
Wanage order >> interface and user click
Manage order button.
5. The application will redirect user to <<
Insert order type id >> interface.
6. The user need to input the id of the order
type that they want to manage and click
"Done" button then system will redirect to
<< Manage order from user>> interface.
7 System will display all the order which is
related to that type of order
related to that type of order.
8. The user can choose order to view and
input comment and click "Comment" button
to update the progress of the order.

	9. Next, it will redirect back to << Manage
	order from user >> interface.
Post-Condition	An order will be available for add, view, delete
	and being updated in the database.

 Table 2. 6: Use case description for make order
 Image: Comparison of the second se

## 2.1.7 Manage Complaint

### **Use Case Diagram**



Figure 2. 8: Use case for make complaint

## Use Case Description

Use Case ID	UMPCA-UC-700
Use Case Name	Manage Complaint

Brief Description	This use case allow user to make complaint and
	the admin can view it and reply on these
	complaint.
Actor	User and admin
Pre – Condition	User had register an account and already login
	to the system.
Basic Flow	User
	1. The use case start when user login to the UMP Confession Application.
	2. The application will display < <user page="">&gt; interface.</user>
	3. User click "My Complaint" button and
	system will redirect to < <my complaint="">&gt;</my>
	interface.
	4. A list of complaint that user made before will
	be retrieve from database and display at this
	page.
Alternative Flow	A1: Add Complaint
	1. Continue at step 4 in Basic flow.
	2. User click "Make Complaint" button.
	3. The application will redirect user to < <make< td=""></make<>
	Complaint>> interface.
	4. User input the content of complaint and click
	"Complaint Now" button.
	A2: Delete Complaint (User)
	1. The use case continue at step 4 in basic flow.

	2. User select compleint that want to delete and
	2. User select complaint that want to delete and
	click "Delete" button.
	A3: Manage complaint (Admin)
	1. After login as admin, admin can click
	"Manage Complaint" button in < <admin< th=""></admin<>
	Page>> Interface and application will redirect
	admin to << Manage Complaint>> interface.
	2. Admin select the complaint that they want to
	reply, and input reply on that complaint.
	3. Admin click "Reply" button and application
	will redirect to < <admin page="">&gt;.</admin>
Post-Condition	The complaint can be manage by the user and
	admin successfully.

 Table 2. 7: Use case description for manage complaint

# 2.2 SEQUENCE DIAGRAM

#### 2.2.1 Manage registration



Figure 2. 9: Manage registration Basic Flow Sequence Diagram

In module manage registration, the user can click sign up button and it will redirect to login interface. Next, controller will display the register interface to the user so the user can input the data that need for register account and click "Register" button. Then, the data of user will store in database and controller will display login interface to the user.





Figure 2. 10 Manage registration [Alternative Flow A1: Having account] Sequence Diagram

In alternative flow A1 of module manage registration, the user who already register account can input their username and password that need for log in and click "Log in" button. Then, the controller will send data to database for verify data and will display login success message to the user if login success.



Figure 2. 11: Manage registration [Alternative Flow A2: Login as admin] Sequence Diagram

In alternative flow A2 of module manage registration which is login as admin, the admin need to click "Login as admin" button to redirect to Login as Admin interface. Next, the admin can input username and password of admin (admin and admin123) that need for log in and click "Log in" button. Then,

the controller will send data to database for verify data and will display login success message to the admin if login success.

#### A3 : Check account (View account)



#### Figure 2. 12: Manage registration

#### [Alternative Flow A3: Check account (View account)] Sequence Diagram

In alternative flow A3 of module manage registration which is view account, the admin can approve the account that made register to become valid user so they can login with their account. After login as admin, the admin can click "Check account" and the controller will send data to database for retrieve data and will display data of user which are still invalid.



A4 : Check account (Approve account)

Figure 2. 13: Manage registration [Alternative Flow A4: Check account (Approve account)] Sequence Diagram

In alternative flow A4 of module manage registration which is check account (Approve account), the admin can approve the account that made register to become valid user so they can login with their account. After login as admin, the admin can click "Check account" and the controller will send data to database for retrieve data and will display data of user which are still invalid. Next, the admin can check on the picture which upload by the user and verify that are the user valid to use the application or not. If admin state that the user is valid for use the application, click "Approve" button and the situation of user which is invalid will be update to valid in database. Next, it will redirect to check account interface.



A5 : Check account (Reject account)

Figure 2. 14: Manage registration [Alternative Flow A5: Check account (Reject account)] Sequence Diagram

In alternative flow A5 of module manage registration which is check account (Reject account), the admin can reject the account that made register to become valid user so they can login with their account. After login as admin, the admin can click "Check account" and the controller will send data to database for retrieve data and will display data of user which are still invalid. Next, the admin can check on the picture which upload by the user and verify that are the user valid to use the application or not. If admin state that the user is invalid for use the application, click "Reject" button and the user will be deleted from the database. Next, it will redirect back to check account interface.



Figure 2. 15: Manage registration [Exception Flow E1: wrong username and password or invalid account] Sequence Diagram

In exception flow E1 of module manage registration, the user will redirect to login interface and a login fail message will be display if wrong username and password or invalid account is given for login.



#### 2.2.2 Manage Post



In module manage post, all post will be display in the User Page interface after the user log in to the application.







In alternative flow A1 of module manage post, the user who already log in into application can click "Make Post" button. Then, the controller will display Make Post interface. The user can select what type of post, input the content of post and upload image for post if they want. Next, user can click "Done" button to store data in Firebase. Lastly, it will redirect to User Page interface.



Figure 2. 18: Manage Post [Alternative Flow A2: Edit Post] Sequence Diagram

In alternative flow A2 of module manage post which is edit post, the user who already login can click "My post" button on User Page interface. Then, the controller will display manage Post interface. The user can select the post that want to edit and click "Edit" button to delete data in Firebase. Lastly, it will redirect to User Page interface.



Figure 2. 19: Manage Post [Alternative Flow A3: Delete Post] Sequence Diagram

In alternative flow A3 of module manage post which is delete post, the user who already login can click "My post" button on User Page interface. Then, the controller will display manage Post interface. The user can select the post that want to delete and click "Delete" button to delete data in Firebase. Lastly, it will redirect to User Page interface.



A4: Delete Post as admin



In alternative flow A4 of module manage post which is delete post as admin, the admin who already login can click "Manage post" button on Admin Page interface. Then, the controller will display Admin Manage Post interface. The admin can select the post that want to delete and click "Delete" button to delete data in Firebase. Lastly, it will redirect to Admin Page interface.



#### A5: View in different type of post

#### Figure 2. 21: Manage Post [Alternative Flow A5: View in different type of post] Sequence Diagram

In alternative flow A5 of module manage post which is view different type of post, user who already login into user main page, the user can click dropdown list menu to select type of post that they want to view. Next, they can click <<Search>> button after select the type of post that user want. Then, all post with the type of post that the user choose will be display.

#### 2.2.3 Manage Comment



Figure 2. 22 : Manage comment Basic Flow Sequence Diagram

In basic flow of module add comment, the user who already login can view all post at User Page interface. Then, user can select a post that they want to give comment. Next, the user can input content of comment and click "Comment" button to store data in Firebase. Lastly, it will redirect to User Page interface.





Figure 2. 23 : Manage announcement Basic Flow Sequence Diagram

In basic flow of module manage announcement, the admin can click "Manage announcement" button and it will redirect to manage announcement interface. Then, the admin can click "Make announcement" button and it will redirect to make announcement interface. Next, the admin can input content of announcement and can click "Done" button to store data in Firebase. Lastly, it will redirect to Admin Page interface.



A1: Edit announcement

#### Figure 2. 24: Manage announcement [Alternative Flow A1: Edit announcement] Sequence Diagram

In alternative flow A1 of module manage announcement which is edit announcement, the admin can click "Manage announcement" button and it will redirect to manage announcement interface and all announcement will be display. Next, the admin can select the announcement that want to edit and click "Edit" button to edit data in Firebase. Lastly, it will redirect to Admin Page interface.

A2: Delete announcement



#### Figure 2. 25: Manage announcement [Alternative Flow A2: Delete announcement] Sequence Diagram

In alternative flow A2 of module manage announcement which is delete announcement, the admin can click "Manage announcement" button and it will redirect to manage announcement interface and all announcement will be display. Next, the admin can select the announcement that want to delete and click "Delete" button to delete data in Firebase. Lastly, it will redirect to Admin Page interface.



#### 2.2.5 Manage order type



In basic flow of manage order type module, admin can click "Manage order type" in admin page interface and it will redirect to manage order type interface. The list of type of order will be display if exist in database. Next, admin click "Add new order type" and controller will display add new order type interface. After input detail of new order type and click "Add" button to store this new order type. Lastly, it will redirect to Admin Page interface.



A1: Delete order type

Figure 2. 27: Manage order type [Alternative Flow A1: Delete order type] Sequence Diagram

In alternative flow A1 of manage order type module which is delete order type, admin can click "Manage order type" in admin page interface and it will redirect to manage order type interface. The list of type of order will be display if exist in database.Next, admin can select order type that want to remove and click "Delete" button to delete this order type from database. Lastly, it will redirect to Admin Page interface.

#### 2.2.6 Manage order



Figure 2. 28: Manage order Basic Flow Sequence Diagram

In basic flow of module manage order, the user can click "My order" button and it will redirect to Manage order interface. The list of order will be display if exist in database. Next, user can click "Add order" and the controller will display add order interface. Next, the user can select type of order that available in database and input details of order. Then the data of order will be store in Firebase after the user click "Done" button. Lastly, it will redirect to User Page interface.



Figure 2. 29: Manage order [Alternative Flow A1: Delete order] Sequence Diagram

In alternative flow A1 of module manage order which is delete order, the user can click "My order" button and it will redirect to Manage order interface. The list of order will be display if exist in database. Next, user can select the order that want to delete and click "Delete" button. Then, the order will be remove from Firebase. Lastly, it will redirect to User Page interface.

A1: Delete order



Figure 2. 30: Manage Order [Alternative Flow A2: View order] Sequence Diagram

In alternative flow A2 of module manage order which is view order, the user can click "My order" button and it will redirect to Manage order interface. The list of order will be display if exist in database. Next, user can select type of order in filter and click done button so they can see the order that only in a specific type of order. Next, the user can select which order that they want to choose and the details of that selected order will be display in view order interface. User can input comment on that order. Then the data of order will be update in Firebase after the user click "Comment" button. Lastly, it will redirect to User Page interface.

SRS UMPCA \_2022\_V1 4



#### 2.2.7 Manage Complaint



In module manage complaint, the user who already log in into application can click "Make Complaint" button. Then, the controller will display <<Make Complaint interface>>. The user can input the content of complaint and click "Complaint Now" button to store data in Firebase. Lastly, it will redirect to User Page interface.



Figure 2. 32: Manage Complaint [Alternative Flow A1: Delete complaint] Sequence Diagram

In alternative flow A1 of module manage complaint which is delete complaint, the user who already log in into application can click "Make Complaint" button. Then, the controller will display <<Make Complaint>> interface. It will display all complaint that the user made at bottom of interface. User can view reply from admin at here. User can click "Delete" button to delete complaint from firebase. Lastly, it will redirect to User Page interface.





In alternative flow A2 of module manage complaint which is reply complaint, the admin who already log in into application can click "Manage Complaint" button. Then, the controller will display Manage Complaint interface. It will display all complaint that all the user made. Admin can give reply on these complaint here and click "Update" button reply and update status of complaint. Lastly, it will redirect to Admin Page interface.

# 3.1 INTERFACE DESIGN

LOGIN UMP Confession Application
Username Password Log In Register first if dont have accout yet Register
Login as admin

Figure 3. 1 : Login interface of UMP Confession Application

UMP Confession Application
Username Password Log In

Figure 3. 2 : LoginAsAdmin interface of UMP Confession Application

Register		
UMP Confession Application		
Username		
Password		
Upload an Image to prove you are from UMP		
Phone number		
Register		

Figure 3. 3: Register interface of UMP Confession Application

USER	USER PAGE		
	User Main Page		
	My Post	My order	
	Make Post	My Complaint	
	Content of announcement		
	VIEW POST		

Figure 3. 4: UserPage interface of UMP Confession Application



Figure 3. 5: AdminPage interface of UMP Confession Application

Check account	шр	
UMP	Confession Applica	ition
User who are still not invalid		
username	Image upload for admin to approve as valid user	Approve Reject
username	Image upload for admin to approve as valid user	Approve Reject

Figure 3. 6: CheckAccount interface of UMP Confession Application

UMP Confession	Application
LIST OF P	OST
Content of post	Edit
Containe or pose	Delete
Content of post	Edit
	Delete

Figure 3. 7: ManagePost interface of UMP Confession Application

UMP Confessi	ion Application	
Make a Post now !		
Type of post:	▼	
Write anytthingd you want :		
Upload image for post :		
Mak	e Post	

Figure 3. 8: MakePost interface of UMP Confession Application
UMP Confession Application
Edit Post ID
Write anytthing you want :
Done

Figure 3. 9: EditPost interface of UMP Confession Application



Figure 3. 10: ViewPost interface of UMP Confession Application

Manage Order				
	Add order	Filter by		Done
	LIST OF ORDER			
	Content of or	rder	Delete	,
	Content of or	rder	Delete	,
Ľ				
L				

Figure 3. 11 : ManageOrder interface of UMP Confession Application



Figure 3. 12: AddOrder interface of UMP Confession Application

View ord	UMP Confession Application
	Order ID
	Content of selected order
Co	mment:
	Comment

Figure 3. 13: ViewOrder interface of UMP Confession Application

Manage Announcement		
Make Announcement		
List of Announcement		
Content of Announcement Edit Delete		

Figure 3. 14: ManageAnnouncement interface of UMP Confession Application



Figure 3. 15: MakeAnnouncement interface of UMP Confession Application



Figure 3. 16: EditAnnouncement interface of UMP Confession Application

Manage order type	Application
Add new order	type
Order type that av	ailable now
CAFE KK5	Delete
HALL RENT	Delete

Figure 3. 17: ManageOrderType interface of UMP Confession Application

Add new order type		
Add new order type		
Type of order:		
Note: Please make sure the apply for add this new order type for user is valid and the people who responsible on this new order type is available.		
Add		

Figure 3. 18: AddOrderType interface of UMP Confession Application

Admin Manage Post
LIST OF POST
Content of post Delete
Content of post Delete

Figure 3. 19: AdminManagePost interface of UMP Confession Application

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FKOM

Manage	Manage complaint		
	Make Complaint		
E	LIST OF Complaint		
	Content of Complaint Status		
	Reply from Admin Delete		
E			

Figure 3. 20: My Complaint interface of UMP Confession Application

Make complaint
Any complaint? Content of Complaint
Complaint now

Figure 3. 21: Make Complaint interface of UMP Confession Application

Manage Complaint
Complaint received from user
content of complaint
Relpy from admin Update
content of complaint
Relpy from admin Update

Figure 3. 22: Manage Complaint interface of UMP Confession Application

U	MP Confession Application
	Post ID
	Content of selected post
Comment:	
	Comment

Figure 3. 23: Add comment interface of UMP Confession Application

## 3.2 HARDWARE AND SOFTWARE SPECIFICATION

Hardware	Specification	Description
Android Mobile device	Redmi Note 10	A device that can used to run the UMP
		Confession Application
Laptop	AMD Ryzen 7 5800H	To run Android Studio which is used
	with Radeon Graphics	to develop UMP Confession
		Application

## Table 3. 1: Hardware Specification

Hardware	Specification	Description
Microsoft Office Word	Version 2013	Software that used for documentation
2013		
Windows 11	Windows 11 Home	To run Android Studio which is used
	Version	to develop UMP Confession
		Application
Android Studio	Version 4.1	An integrated development
		environment designed for Android
		application development using Java
		language.
Firebase	Version 20.2.3	A mobile and web application
		development platform that enable real
		time syncing of data

Draw.io	Web version	Used to design and create all the
		related diagrams in the
		documentations.
Figma	Web version	Used as a tool to develop the design prototype of UMP Confession Application.

Table 3.2: Software Specification

APPENDIX B SDD



# SOFTWARE DESIGN DESCRIPTION (SDD)

[UMP Confession Application]

## DOCUMENT APPROVAL

	Name	Date
Authenticated by:		
Name		
Approved by:		
Client		

Archiving Place :

## TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	ii
TABLE OF CONTENT	iii
LIST OF FIGURES	iv
No table of figures entries found.No table of figures entries found. vi	LIST OF TABLES
LIST OF APPENDICES	X
1.1 PROJECT DESCRIPTION	1
1.2 SYSTEM IDENTIFICATION	1
1.3 ARCHITECTURE / BLUE PRINT	2
1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION	3
2.1 DETAILED DESCRIPTION	
2.2 DATA DICTIONARY	76

## LIST OF FIGURES

Figure 1. 1: Architecture of UMP Confession Application
Figure 1. 2: Login interface of UMP Confession Application
Figure 1. 3: LoginAsAdmin interface of UMP Confession Application
Figure 1. 4: Register interface of UMP Confession Application
Figure 1. 5: UserPage interface of UMP Confession Application
Figure 1. 6: AdminPage interface of UMP Confession Application
Figure 1. 7: CheckAccount interface of UMP Confession Application
Figure 1. 8: ManagePost interface of UMP Confession Application11
Figure 1. 9: MakePost interface of UMP Confession Application11
Figure 1. 10: EditPost interface of UMP Confession Application
Figure 1. 11: ViewPost interface of UMP Confession Application
Figure 1. 12: ManageOrder interface of UMP Confession Application
Figure 1. 13: AddOrder interface of UMP Confession Application
Figure 1. 14: ViewOrder interface of UMP Confession Application
Figure 1.15: ManageAnnouncement interface of UMP Confession Application14
Figure 1. 16: MakeAnnouncement interface of UMP Confession Application15
Figure 1. 17: EditAnnouncement interface of UMP Confession Application
Figure 1. 18: ManageOrderType interface of UMP Confession Application
Figure 1. 19: AddOrderType interface of UMP Confession Application
Figure 1. 20: AdminManagePost interface of UMP Confession Application
SDD UMPCA _2022_V1 iv

Figure 1. 21:	MakeComplaint interface of UMP Confession Application	17
Figure 1. 22:	AdminManagePost interface of UMP Confession Application	17
Figure 1. 23:	Login controller of UMP Confession Application	18
Figure 1. 24:	LoginAsAdmin controller of UMP Confession Application	18
Figure 1. 25:	Register controller of UMP Confession Application	19
Figure 1. 26:	UserPage controller of UMP Confession Application	20
Figure 1. 27:	AdminPage controller of UMP Confession Application	20
Figure 1. 28:	CheckAccount controller of UMP Confession Application	21
Figure 1. 29:	ManagePost controller of UMP Confession Application	21
Figure 1. 30:	MakePost controller of UMP Confession Application	22
Figure 1. 31:	EditPost controller of UMP Confession Application	23
Figure 1. 32:	ViewPost controller of UMP Confession Application	23
Figure 1. 33:	ManageOrder controller of UMP Confession Application	24
Figure 1. 34:	AddOrder controller of UMP Confession Application	24
Figure 1. 35:	ViewOrder controller of UMP Confession Application	25
Figure 1. 36:	ManageAnnouncement controller of UMP Confession Application	25
Figure 1. 37:	MakeAnnouncement controller of UMP Confession Application	26
Figure 1. 38:	EditAnnouncement controller of UMP Confession Application	26
Figure 1. 39:	ManageOrderType controller of UMP Confession Application	27
Figure 1. 40:	AddOrderType controller of UMP Confession Application	27
Figure 1. 41:	AdminManagePost controller of UMP Confession Application	28
	SDD UMPCA _2022_V1	/

Figure 1. 42:	MakeComplaint controller of UMP Confession Application	28
Figure 1. 43:	Manage Complaint controller of UMP Confession Application	29
Figure 1. 44:	User model of UMP Confession Application	29
Figure 1. 45:	Admin model of UMP Confession Application	30
Figure 1. 46:	Post model of UMP Confession Application	30
Figure 1. 47:	Comment model of UMP Confession Application	30
Figure 1. 48:	Order model of UMP Confession Application	31
Figure 1. 49:	Announcement model of UMP Confession Application	31
Figure 1. 50:	Order type model of UMP Confession Application	31
Figure 1. 51:	Complaint model of UMP Confession Application	31

## LIST OF TABLES

Table 1.1 V	View layer of MVC	4
Table 1.2:	Controller layer of MVC	6
Table 1.3:	Model layer of MVC	7

Table 2. 1 : Login interface of UMP Confession Application    3	32
Table 2. 2: LoginAsAdmin interface of UMP Confession Application	33
Table 2. 3: Register interface of UMP Confession Application	34
Table 2. 4: UserPage interface of UMP Confession Application	35
Table 2. 5: AdminPage interface of UMP Confession Application	36
Table 2. 6: CheckAccount interface of UMP Confession Application	37
Table 2. 7: ManagePost interface of UMP Confession Application	38
Table 2. 8: MakePost interface of UMP Confession Application	39
Table 2. 9: EditPost interface of UMP Confession Application4	10
Table 2. 10: ViewPost interface of UMP Confession Application4	11
Table 2. 11: ManageOrder interface of UMP Confession Application4	11
Table 2. 12: AddOrder interface of UMP Confession Application4	12
Table 2. 13: ViewOrder interface of UMP Confession Application4	13
Table 2. 14: ManageAnnouncement interface of UMP Confession Application	14
Table 2. 15: MakeAnnouncement interface of UMP Confession Application	14
Table 2. 16: EditAnnouncement interface of UMP Confession Application4	<del>1</del> 5
Table 2. 17: ManageOrderType interface of UMP Confession Application	15
Table 2. 18: AddOrderType interface of UMP Confession Application	16
Table 2. 19: AdminManagePost interface of UMP Confession Application	16
Table 2. 20: MakeComplaint interface of UMP Confession Application	17

Table 2. 21: ManageComplaint interface of UMP Confession Application	47
Table 2. 22: Login controller of UMP Confession Application	49
Table 2. 23: LoginAsAdmin controller of UMP Confession Application	50
Table 2. 24: Register controller of UMP Confession Application	51
Table 2. 25: UserPage controller of UMP Confession Application	53
Table 2. 26: AdminPage controller of UMP Confession Application	55
Table 2. 27: CheckAccount controller of UMP Confession Application	56
Table 2. 28: ManagePost controller of UMP Confession Application	57
Table 2. 29:    MakePost controller of UMP Confession Application	58
Table 2. 30: EditPost controller of UMP Confession Application	59
Table 2. 31: ViewPost controller of UMP Confession Application	60
Table 2. 32: ManageOrder controller of UMP Confession Application	61
Table 2. 33: AddOrder controller of UMP Confession Application	62
Table 2. 34: ViewOrder controller of UMP Confession Application	63
Table 2. 35:       ManageAnnouncement controller of UMP Confession Application	64
Table 2. 36:       MakeAnnouncement controller of UMP Confession Application	65
Table 2. 37: EditAnnouncement controller of UMP Confession Application	66
Table 2. 38: ManageOrderType controller of UMP Confession Application	67
Table 2. 39:       AddOrderType controller of UMP Confession Application	68
Table 2. 40:    AdminManagePost controller of UMP Confession Application	69
Table 2. 41: MakeComplaint controller of UMP Confession Application	70
Table 2. 42: ManageComplaint controller of UMP Confession Application	71

Table 2. 43:	User model of UMP Confession Application	72
Table 2. 44:	Admin model of UMP Confession Application	72
Table 2. 45:	Post model of UMP Confession Application	73
Table 2. 46:	Comment model of UMP Confession Application	73
Table 2. 47:	Order model of UMP Confession Application	74
Table 2. 48:	Announcement model of UMP Confession Application	74
Table 2. 49:	Order type model of UMP Confession Application	75
Table 2. 50:	Complaint model of UMP Confession Application	75
Table 2. 51:	Data Dictionary for User	76
Table 2. 52:	Data Dictionary for Admin	76
Table 2. 53:	Data Dictionary for Post	76
Table 2. 54:	Data Dictionary for Comment	77
Table 2. 55:	Data Dictionary for Announcement	77
Table 2. 56:	Data Dictionary for Order	77
T-1-1- 0 57.		
Table 2.57.	Data Dictionary for Order type	78

## LIST OF APPENDICES

#### **CHAPTER 1**

#### **1.1 PROJECT DESCRIPTION**

UMP Confession application is an application that is develop for the social use of the people in a University and this UMP Confession application is aim for use by the student or anyone who stay in Universiti Malaysia Pahang. It will consist of 6 modules which are manage registration, manage post, add comment, manage announcement, manage order type and manage order.

In the manage registration module, user able to register for a new account so they can log into the application. Since the user need to register with an evidence that can prove that they are stay in UMP, admin is able to view all user details and remove user from the application or set user to be valid for using the application.

In manage post, user can make post by select the type of post that they want to post such as confession or complaint. Besides, they also manage to edit and delete their own post. Admin also have permission to delete any post if they want. The post that make will not showing the username. In add comment, the user can give comment to any post if they want. In manage announcement, the admin can make announcement to let everyone view the announcement since the announcement will be on the top of the main page of user. Beside, admin is free to edit them or remove them from database.

In manage order and manage order type, it will be more complicated and these two module is related to each other. This application only provide a platform for the user who wish to make order and other user who need to receive the order. For example, the owner of badminton court can find and ask the admin to add a new order type so the order type of rent badminton court will be available for the user to choose. Then, other user can make order to rent for badminton court. Besides, the user who own the badminton court can view order from the application and leave comment to reply to the user who make order.

### **1.2 SYSTEM IDENTIFICATION**

System Identification Number: UMPCA \_2022\_V1

System Name: UMP Confession Application

System Abbreviation: UMPCA

Year: 2022

Version: V1



## **1.3 ARCHITECTURE / BLUE PRINT**

Figure 1.1: Architecture of UMP Confession Application

# 1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

## View Layer

Class	Description
Login interface	The interface for user to login into the
	application.
Login as Admin interface	The interface for admin to login into the
	application.
Register interface	The interface for user to register an account.
User Page interface	The interface of main page for the user and all
	function of the application can be used in this
	interface.
Admin Page interface	The interface of main page for the admin and all
	function of the application can be used in this
	interface.
Check account interface	The interface for admin to check all the account
	that register by the user which are invalid.
Manage Post interface	The interface for user and admin manage all the
	post.
Make Post interface	The interface for user to make post by filling a
	form.
Edit Post interface	The interface for user to edit the post that they
	make.
View Post interface	The interface for user and admin to view a
	specific post and add or delete comment on the
	post.
Manage Oder interface	The interface for user to manage order.

Add Order interface	The interface for user to add a new order by
	filling a form.
View Order interface	The interface for user to view the order that
	available in database and add comment to
	undate the status of the order
	update the status of the order.
Manage Announcement interface	The interface for admin to manage the
	announcement
	amouncement.
Make Announcement interface	The interface for admin to make a new
	announcement
Edit Announcement interface	The interface for admin to edit announcement.
Manage order type interface	The interface for user and admin to manage
	order type that will available for user to choose
	as type to make order
	as type to make order.
Add order type interface	The interface for admin to add order type so it
	will be available for user to choose as type to
	maka order
	make order.
Admin Manage Post interface	The interface for admin to manage the post that
	make by other user
	hake by other user.
Make Complaint interface	The interface for user to manage their own
	complaint
Manage Complaint interface	The interface for admin to manage all
	complaint.

 Table 1. 1 View layer of MVC

# **Controller layer**

Class	Description
Login controller	The controller for user and admin to login into
	the application.
Login as Admin controller	The controller for admin to login into the
	application.
Register controller	The controller for user to register an account.
User Page controller	The controller of main page for the user and all
	function of the application can be used in this
	interface.
Admin Page controller	The controller of main page for the admin and
	all function of the application can be used in this
	interface.
Check account controller	The controller for admin to check all the
	account that register by the user which are
	invalid.
Manage Post controller	The controller for user and admin manage all
	the post.
Make Post controller	The controller for user to make post by filling a
	form.
Edit Post controller	The controller for user to edit the post that they
	make.
View Post controller	The controller for user and admin to view a
	specific post and add or delete comment on the
	post.
Manage order controller	The controller for user to manage order.
Add order controller	The controller for user to add a new order by
	filling a form.

	-
View order controller	The controller for user to view the order that
	available in database and leave comment to
	update the status of the order.
Manage Announcement controller	The controller for admin to manage the
	announcement.
Make Announcement controller	The controller for admin to make a new
	announcement.
Edit Announcement controller	The controller for admin to edit announcement.
Managa anden tama agentuallan	The interface for more and a durin to more an
Manage order type controller	The internace for user and admin to manage
	order type that will available for user to choose
	as type to make order.
Add order type controller	The controller for admin to add order type so it
	will be available for user to choose as type to
	make order.
Admin Manage Post controller	The controller for admin to manage the post that
	make by other user.
Make Complaint controller	The controller for user to manage their own
	complaint.
Manage Complaint controller	The controller for admin to manage all
	complaint.

 Table 1. 2: Controller layer of MVC
 Image: Controller layer of MVC

# Model layer

Class	Description
User	This model is used to store and retrieve data of
	user.
Admin	This model is used to store and retrieve data of
	admin.
Post	This model is used to store and retrieve data of
	post.
Comment	This model is used to store and retrieve data of
	comment.
Announcement	This model is used to store and retrieve data of
	announcement.
Order type	This model is used to store and retrieve data of
	order type.
Order	This model is used to store and retrieve data of
	order.
Complaint	This model is used to store and retrieve data of
	complaint.

 Table 1.3: Model layer of MVC



Figure 1. 2: Login interface of UMP Confession Application



Figure 1.3: LoginAsAdmin interface of UMP Confession Application
Register				
Register Interface				
-ID: INT				
-username: String				
-password: String				
-HP: String				
-validornot: INT				
+image: Image				

Figure 1. 4: Register interface of UMP Confession Application



Figure 1.5: UserPage interface of UMP Confession Application



Figure 1. 6: AdminPage interface of UMP Confession Application

(	Check account		
,			
	CheckAccount Interface		
	-ID: INT		
-username: String			
	-password: String		
	-HP: String		
	+image: Image		
	-validornot: INT		

Figure 1.7: CheckAccount interface of UMP Confession Application

Manage Post		
ManagePost Interface		
-Post_ID: INT		
+ContentOfPost: String		
+IypeOtPost: String		
-ID: INT		

Figure 1.8: ManagePost interface of UMP Confession Application

Make post	
MakePost Interface	
+ContentOfPost: String	
+TypeOfPost: String	
+ImgofPost: IMAGE	
-10. 111	

Figure 1. 9: MakePost interface of UMP Confession Application



Figure 1. 10: EditPost interface of UMP Confession Application



Figure 1. 11: ViewPost interface of UMP Confession Application

Ν	/lanage order				
	ManageOrder Interface				
	-Order_ID: INT				
	+DetailOfOrder: String +Comment: String				
	+Order_Type_ID:INT				
	+ID:INT				

Figure 1. 12: ManageOrder interface of UMP Confession Application

	Add order			
	AddOrder Interface			
	-Order_ID: INT			
	+DetailOfOrder: String			
	+Comment: String			
	+Order_Type_ID:INT			
	+ID:INT			

Figure 1. 13: AddOrder interface of UMP Confession Application

View order				
ViewOrder Interface				
-Order_ID: INT				
+DetailOfOrder: String				
+Comment: String				
+Order_Type_ID:INT				
+ID:INT				

Figure 1. 14: ViewOrder interface of UMP Confession Application



Figure 1. 15: ManageAnnouncement interface of UMP Confession Application



Figure 1. 16: MakeAnnouncement interface of UMP Confession Application



Figure 1. 17: EditAnnouncement interface of UMP Confession Application

Manage order type					
	ManageOrderType Interface				
	+Order_Type_ID:INT				
	+OrderType: String				

Figure 1. 18: ManageOrderType interface of UMP Confession Application

	Add order type					
	·					
	AddOrderType Interface					
	+Order_Type_ID:INT					
	+OrderType: String					

Figure 1. 19: AddOrderType interface of UMP Confession Application

Admin Manage Post			
	AdminManagePost Interface		
	-Post_ID: INT		
+ContentOfPost: String			
	-ID: INT		

Figure 1. 20: AdminManagePost interface of UMP Confession Application



Figure 1. 21: MakeComplaint interface of UMP Confession Application

Manage Complaint				
	+ContentOfComplaint: String			
	+Status: String			
	+Reply: String			

Figure 1. 22: AdminManagePost interface of UMP Confession Application



Figure 1. 23: Login controller of UMP Confession Application



Figure 1. 24: LoginAsAdmin controller of UMP Confession Application



Figure 1. 25: Register controller of UMP Confession Application

User Page	
User Page	Controller
-ID: INT	
-username: St	ring
-password: St	ring
+image: Image	,
-validornot: IN	г
-HP: String	
- usermainpag + viewType()	e():void
L	

Figure 1. 26: UserPage controller of UMP Confession Application



Figure 1. 27: AdminPage controller of UMP Confession Application

Check account			
·			
Check account Controller			
-ID: INT			
-username: String			
-password: String			
-HP: String			
+image: Image			
-validornot: INT			
- checkUser():void			

Figure 1. 28: CheckAccount controller of UMP Confession Application

Manage Post			
Manage post Co	ontroller		
-Post_ID: INT			
+TypeOfPost: String	+TypeOfPost: String		
+ContentOfPost: St	tring		
+ImgofPost: IMAGE	+ImgofPost: IMAGE		
-ID: INT			
# managePost():voi	id		

Figure 1. 29: ManagePost controller of UMP Confession Application

Make post	
Make post Controller	
-Post_ID: INT	
+TypeOfPost: String	
+ContentOfPost: String	
+ImgotPost: IMAGE	
+makePost():void	

Figure 1. 30: MakePost controller of UMP Confession Application



Figure 1. 31: EditPost controller of UMP Confession Application



Figure 1. 32: ViewPost controller of UMP Confession Application



Figure 1. 33: ManageOrder controller of UMP Confession Application



Figure 1. 34: AddOrder controller of UMP Confession Application



Figure 1. 35: ViewOrder controller of UMP Confession Application



Figure 1. 36: ManageAnnouncement controller of UMP Confession Application



Figure 1. 37: MakeAnnouncement controller of UMP Confession Application



Figure 1. 38: EditAnnouncement controller of UMP Confession Application



Figure 1. 39: ManageOrderType controller of UMP Confession Application

Add order type Controller +Order_Type_ID:INT	
Add order type Controller +Order_Type_ID:INT	
+Order_Type_ID:INT	
+Order_Type_ID:INT	
+OrderType: String # addOrderType():void	

Figure 1. 40: AddOrderType controller of UMP Confession Application

Login as Admin Controller		
-ID: INT		
-username: String		
-password: String		
- login():void		

Figure 1. 41: AdminManagePost controller of UMP Confession Application



Figure 1. 42: MakeComplaint controller of UMP Confession Application



Figure 1. 43: Manage Complaint controller of UMP Confession Application



Figure 1. 44: User model of UMP Confession Application



Figure 1. 45: Admin model of UMP Confession Application



Figure 1. 46: Post model of UMP Confession Application



Figure 1. 47: Comment model of UMP Confession Application



Figure 1. 48: Order model of UMP Confession Application



Figure 1. 49: Announcement model of UMP Confession Application



Figure 1. 50: Order type model of UMP Confession Application



Figure 1. 51: Complaint model of UMP Confession Application

## 2.1 DETAILED DESCRIPTION

Class Type	Boundary Class	
Responsibility	This interface is used to login by user.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	validornot	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	·

Table 2. 1 : Login interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for admin to login.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 2: LoginAsAdmin interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for registration	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	НР	String
	Image	IMAGE
	validornot	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 3: Register interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is user main page.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	НР	String
	Image	IMAGE
	validornot	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 4: UserPage interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is admin main page.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	<u> </u>

Table 2. 5: AdminPage interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is to for admin to approve or reject account to be valid.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	НР	String
	Image	IMAGE
	validornot	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 6: CheckAccount interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to manage post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ImgofPost	IMAGE
	ContentOfPost	String
	ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

 Table 2. 7: ManagePost interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to add post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ImgofPost	IMAGE
	ContentOfPost	String
	ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 8: MakePost interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to edit their post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ImgofPost	IMAGE
	ContentOfPost	String
	ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 9: EditPost interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to view post and add comment to any post.	
Attributes	Attributes Name	Attributes Type
	Comment_ID	INT
	ContentOfComment	String
	Post_ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 10: ViewPost interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is to for user to manage order.	
Attributes	Attributes Name	Attributes Type
	Order_ID	INT
	Order_Type_ID	INT
	DetailsOfOrder	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

## Table 2. 11: ManageOrder interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to add order.	
Attributes	Attributes Name	Attributes Type
	Order_ID	INT
	DetailsOfOrder	String
	Comment	String
	Order_Type_ID	INT
	ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 12: AddOrder interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for user to view order and add comment to the order	
	for update the progress of order.	
Attributes	Attributes Name	Attributes Type
	Order_ID	INT
	DetailsOfOrder	String
	Comment	String
	Order_Type_ID	INT
	ID	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 13: ViewOrder interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for admin to manage announcement.	
Attributes	Attributes Name	Attributes Type
	Announcement_ID	INT
	ContentOfAnnounce	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 14: ManageAnnouncement interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for admin to add a new announcement.	
Attributes	Attributes Name	Attributes Type
	Announcement_ID	INT
	ContentOfAnnounce	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

 Table 2. 15: MakeAnnouncement interface of UMP Confession Application
Class Type	Boundary Class		
Responsibility	This interface is for admin to edit announcement.		
Attributes	Attributes Name     Attributes Type		
	Announcement_ID	INT	
	ContentOfAnnounce	String	
Methods	Method Name	Description	
	N/A	N/A	
Algorithm	N/A		

# Table 2. 16: EditAnnouncement interface of UMP Confession Application

Class Type	Boundary Class		
Responsibility	This interface is for manage order type.		
Attributes	Attributes Name     Attributes Type		
	Order_Type_ID	INT	
	OrderType	String	
Methods	Method Name	Description	
	N/A	N/A	
Algorithm	N/A	·	

# Table 2. 17: ManageOrderType interface of UMP Confession Application

Class Type	Boundary Class	
Responsibility	This interface is for add new order type.	
Attributes	Attributes Name     Attributes Type	
	Order_Type_ID	INT
	OrderType	String
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 18: AddOrderType interface of UMP Confession Application

Class Type	Boundary Class		
Responsibility	This interface for admin to manage post.		
Attributes	Attributes Name	Attributes Type	
	Post_ID	INT	
	TypeOfPost	String	
	ContentOfPost	String	
	ID	INT	
Methods	Method Name	Description	
	N/A	N/A	
Algorithm	N/A		

Table 2. 19: AdminManagePost interface of UMP Confession Application

Class Type	Boundary Class		
Responsibility	This interface for user to make complaint.		
Attributes	Attributes Name	Attributes Type	
	Complaint_ID	INT	
	ContentOfComplaint	String	
	Status	String	
	Reply	INT	
Methods	Method Name	Description	
	N/A	N/A	
Algorithm	N/A		

Table 2 20.	MakeComplai	nt interface o	of IIMP (	Confession	Application
1 4010 2. 20.	manecomptai	<i>n merjace</i> og	<i>y</i> 0 <i>m</i> 1 <b>v</b>	conjession	- pp nounon

Class Type	Boundary Class	
Responsibility	This interface for admin to manage complaint.	
Attributes	Attributes Name	Attributes Type
	Complaint_ID	INT
	ContentOfComplaint	String
	Status	String
	Reply	INT
Methods	Method Name	Description
	N/A	N/A
Algorithm	N/A	·

Table 2. 21: ManageComplaint interface of UMP Confession Application

Class Type	Controller Class		
Responsibility	This controller is to receive and process login for user.		
Attributes	Attributes Name	Attributes Type	
	ID	INT	
	username	String	
	password	String	
	validornot	String	
Methods	Method Name	Description	
	Login()	To login with an account.	
	onStart()	To check user valid to login or not.	
Algorithm	Login()		
	START IF (click="Register") THEN Call create() method in RegisterController() END IF ENTER username		
	ENTER password		
	IF (click="Login") THEN Call onStart() method in LoginController()		

IF (username==username &&
password==password&&validornot=1)
Redirect to UserPage interface
END IF
END IF
IF (click="Login as admin") THEN
Redirect to Admin interface
END

Table 2. 22: Login controller of UMP Confession Application

Class Type	Controller Class		
Responsibility	This controller is to receive and process login for admin.		
Attributes	Attributes Name	Attributes Type	
	ID	INT	
	username	String	
	password	String	
Methods	Method Name	Description	
	Login()	To login with an account.	
Algorithm	Login()		
	START		
	ENTER username ENTER password IF (click="Login") THEN IF (username==admin && password==admin123) Redirect to AdminPage interface END IF END IF		
	END		

Table 2. 23: LoginAsAdmin controller of UMP Confession Application

Class Type	Controller Class		
Responsibility	This controller is to receive and process data for registration		
Attributes	Attributes Name     Attributes Type		
	ID	INT	
	username	String	
	password	String	
	НР	String	
	Image	Image	
	validornot	INT	
Methods	Method Name	Description	
	CreateUser()	To add a new account into application.	
Algorithm	CreateUser()		
	START		
	READ username		
	READ password		
	READ IMAGE		
	READ HP		
	INSERT INTO "User", store ID, username, password, HP. Image, validornot redirect to Login interface		
	END		

### Table 2. 24: Register controller of UMP Confession Application

Class Type	Controller Class		
Responsibility	This controller is to receive and process data for user main page.		
Attributes	Attributes Name     Attributes Type		
	ID	INT	
	username	String	
	password	String	
	НР	String	
	Image	Image	
	Validornot	INT	
Methods	Method Name	Description	
	usermainpage()	The main page of user after login	
		success.	
	ViewType()	The method used to view post by	
		different type of post.	
Algorithm	usermainpage()		
	START IF (click=MyPost) at UserPage interface Redirect to ManagePost interface ELSE IF (click=MyOrder) at UserPage interface Redirect to ManageOrder interface		

ELSE IF (click=MakePost) at UserPage interface
Redirect to MakePost interface
ELSE IF (click=MakeComplaint) at UserPage interface
Redirect to MakeComplaint interface
END IF
DISPLAY ContentOfAnnouce
DISPLAY ContentOfPost
END
ViewType()
READ TypeOfPost
IF (click=Search) at UserPage interface
DISPLAY ContentOfPost FROM Post WHERE TypeOfPost =
TypeOfPost
END IF
END

Table 2. 25: UserPage controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for admin main page.	
Attributes	Attributes Name     Attributes Type	
	ID	INT
	username	String
	password	String
Methods	Method Name	Description
	adminmainpage ()	The main page of admin after login success.
Algorithm	adminmainpage () START	
	IF (click=ManageAnnoucement) at AdminPage interface	
	Redirect to ManageAnnoucement interfaceELSE IF (click=Managepost) at AdminPage interfaceRedirect to AdminManagePost interfaceELSE IF (click=Checkaccount) at AdminPage interfaceRedirect to CheckAccount interfaceELSE IF (click=Manageordertype) at AdminPage interfaceRedirect to ManageOrdertype interfaceELSE IF (click=ManageComplaint) at AdminPage interface	
	Redirect to ManageCo	mplaint interface
	END IF	
	END	

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for approve or reject account to be valid.	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	НР	String
	Image	String
	validornot	INT
Methods	Method Name	Description
	CheckUser()	To set the user account valid or not.
Algorithm	CheckUser () START DISPLAY username	
	DISPLAY IMAGE FROM "User" WHERE validornot = 0	
	IF (click=Approve) at CheckAccount interface	
	SET validornot = 1	
	ELSE IF (click=Reject) a	t CheckAccount interface
	DELETE FROM User W	HERE username = username

 Table 2. 26: AdminPage controller of UMP Confession Application

END IF
END

Table 2. 27: CheckAccount controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for manage post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ContentOfPost	String
	ID	INT
Methods	Method Name	Description
	managePost()	The page of user to manage post.
Algorithm	managePost () START	
DISPLAY ContentOfPost FROM Post WHERE ID = ID		<b>I</b> Post WHERE ID = ID
	IF (click= ContentOfPost) at ManagePost interface	
	Redirect to ViewPost interface	
	ELSE IF (click=Edit) at ManagePost interface	

Redirect to EditPost interface
ELSE IF (click=Delete) at ManagePost interface
DELETE FROM Post WHERE ContentOfPost = ContentOfPost
END IF
END

Table 2. 28: ManagePost controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for add post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ContentOfPost	String
	ImgofPost	IMAGE
	ID	INT
Methods	Method Name	Description
	makePost()	The method for user to make a
		new post.
Algorithm	makePost ()	
	START	
	READ TypeOfPost	

READ ContentOfPost
READ ImgofPost
INSERT INTO "Post", store Post_ID, TypeOfPost,ContentOfPost,
ImgofPost, ID
Redirect to UserPage interface.
END

 Table 2. 29:
 MakePost controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data to edit user's post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ContentOfPost	String
	ImgofPost	IMAGE
	ID	INT
Methods	Method Name	Description
	editPost ()	The method for user to edit a new
		post.
Algorithm	editPost ()	
	START	
	READ TypeOfPost	

READ ContentOfPost
READ ImgofPost
UPDATE INTO "Post" WHERE Post_ID= "Post_ID"
Redirect to UserPage interface.
END

 Table 2. 30: EditPost controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for user to view post	
	and add comment to a	ny post.
Attributes	Attributes Name	Attributes Type
	Comment_ID	INT
	ContentOfComment	String
	Post_ID	INT
Methods	Method Name	Description
	ViewPost ()	The method for user to view a post.
	addComment()	The method for user to add comment to post.
	DeleteComment()	The method for user to delete comment on
		post.
Algorithm	ViewPost ()	1
	START	
	DISPLAY ContentOf	Post FROM "Post" WHERE Post_ID = Post_ID

DISPLAY Comment_ID FROM "Comment" WHERE Post_ID = Post_ID END addComment() START READ ContentOfComment INSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_ID Redirect to UserPage interface. DeleteComment() READ ContentOfComment IF (click=Delete) DELETE FROM Comment WHERE Comment_ID = Comment_ID END IF END	
Post_IDENDaddComment()STARTREAD ContentOfCommentINSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	DISPLAY Comment_ID FROM "Comment" WHERE Post_ID =
ENDaddComment()STARTREAD ContentOfCommentNSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IF	Post_ID
addComment()STARTREAD ContentOfCommentINSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	END
STARTREAD ContentOfCommentINSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	addComment()
READ ContentOfCommentINSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	START
INSERT INTO "Comment", store Comment_ID, ContentOfComment, Post_IDRedirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	READ ContentOfComment
ContentOfComment, Post_ID Redirect to UserPage interface. DeleteComment() READ ContentOfComment IF (click=Delete) DELETE FROM Comment WHERE Comment_ID = Comment_ID END IF END	INSERT INTO "Comment", store Comment_ID,
Redirect to UserPage interface.DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	ContentOfComment, Post_ID
DeleteComment()READ ContentOfCommentIF (click=Delete)DELETE FROM Comment WHERE Comment_ID = Comment_IDEND IFEND	Redirect to UserPage interface.
READ ContentOfComment IF (click=Delete) DELETE FROM Comment WHERE Comment_ID = Comment_ID END IF END	DeleteComment()
IF (click=Delete) DELETE FROM Comment WHERE Comment_ID = Comment_ID END IF END	READ ContentOfComment
DELETE FROM Comment WHERE Comment_ID = Comment_ID END IF END	IF (click=Delete)
END IF END	<b>DELETE FROM Comment WHERE Comment_ID = Comment_ID</b>
END	END IF
	END

Table 2. 31: ViewPost controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for manage order.	
Attributes	Attributes Name Attributes Type	
	Order_ID	INT
	Order_Type_ID	INT
	DetailsOfOrder	String

Methods	Method Name	Description
	manageOrder()	The page of user to manage order.
Algorithm	manageOrder ()	
	START	
	DISPLAY DetailsOfOrder FROM Order	
	IF (click= Addorder) at ManageOrder interface	
	Redirect to AddOrder interface	
	ELSE IF (click=Delete) at ManageOrder interface	
	DELETE FROM Order WHERE DetailsOfOrder = DetailsOfOrder	
	END IF	
	END	

 Table 2. 32: ManageOrder controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for add order.	
Attributes	Attributes Name	Attributes Type
	Order_ID	INT
	DetailsOfOrder	String
	Comment	String
	Order_Type_ID	INT
	ID	INT

Methods	Method Name	Description
	addOrder()	The method for user to add a new
		order.
Algorithm	addOrder()	
	START	
	READ Order_Type_ID	
	READ DetailsOfOrder	
	INSERT INTO "Order" , store Order_ID, DetailsOfOrder Comment, Order_Type_ID, ID	
	Redirect to UserPage interface.	
	END	

Table 2. 33: AddOrder controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for user to view order and add comment to the order for update the progress of order.	
Attributes	Attributes Name	Attributes Type
	Order_ID	INT
	DetailsOfOrder	String
	Comment	String
	Order_Type_ID	INT

	ID	INT
Methods	Method Name	Description
	viewOrder ()	The method for user to view a order
	addComment ()	The method for user to add a comment to a order.
Algorithm	ViewPost ()	
	START	
	DISPLAY DetailsOfOrder FROM "Order" WHERE Ord	
	Order _ID	
	END	
	addComment()	
	START	
	READ Comment	
	UPDATE INTO "Order"	
	Redirect to UserPage interface.	
	END	

 Table 2. 34: ViewOrder controller of UMP Confession Application

Class Type	Controller Class
Responsibility	This controller is to receive and process data for manage announcement.

SOFTWARE DESIGN DESCRIPTION (SDD)

FKOM

		4 · · · ••• · · · · · · · · · · · · · ·	
Attributes	Attributes Name	Attributes Type	
	Announcement _ID	INT	
	ContentOfAnnouce	String	
Methods	Method Name	Description	
	manageAnnouncement ()	The method for admin to manage	
		announcement.	
Algorithm	manageAnnouncement ()		
	START		
	DISPLAY ContentOfAnnouce FROM Annoucement		
	IF (click= Make Announcement) at ManageAnnouncement interface		
	Redirect to MakeAnnouncement interface ELSE IF (click=Edit) at ManageAnnouncement interface		
	Redirect to EditAnnouncement interface         ELSE IF (click=Delete) at ManageAnnouncement interface         DELETE FROM AnnoucementWHERE ContentOfAnnouce =         ContentOfAnnouce		
	END IF		
	END		

# Table 2. 35: ManageAnnouncement controller of UMP Confession Application

Class Type	Controller Class
Responsibility	This controller is to receive and process data for add announcement.

FKOM

Attributes	Attributes Name	Attributes Type
	Announcement _ID	INT
	ContentOfAnnouce	String
Methods	Method Name	Description
	makeAnnouncement ()	The method for user to add a new
		announcement.
Algorithm	makeAnnouncement ()	
	START	
READ Announcement _ID READ ContentOfAnnounce		
	ContentOfAnnounce	
	Redirect to AdminPage interface.	
	END	

### Table 2. 36: MakeAnnouncement controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for user to edit announcement.	
Attributes	Attributes Name	Attributes Type
	Announcement _ID	INT

FKOM

	ContentOfAnnouce	String
Methods	Method Name	Description
	UpdateAnnouce ()	The method for user to edit
		announcement.
Algorithm	gorithm UpdateAnnouce () START READ ContentOfAnnounce UPDATE INTO "Announcement" WHERE Announcement _ID = "Announcement _ID" Redirect to AdminPage interface.	
	END	

Table 2. 37: EditAnnouncement controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for manage order type.	
Attributes	Attributes Name	Attributes Type
	Order_Type_ID	INT
	OrderType	String

Methods	Method Name	Description
	manageOrderType()	The method for admin to manage
		order type.
Algorithm	manageOrderType ()	
	START         DISPLAY OrderType FROM Order type         IF (click=Delete) at ManageOrderType interface         DELETE FROM Order type WHERE OrderType = OrderType	
	END IF	
	END	

Table 2. 38: ManageOrderType controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for add new order type.	
Attributes	Attributes Name	Attributes Type
	Order_Type_ID	INT
	OrderType	String
Methods	Method Name	Description
	addOrderType ()	The method for user to add a new order type.
Algorithm     addOrderType ()		
	START	

READ OrderType
INSERT INTO "Order Type", store Order_Type_ID, OrderType
Redirect to AdminPage interface.
END

Table 2. 39: AddOrderType controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for admin to manage post.	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ContentOfPost	String
	ID	INT
Methods	Method Name	Description
	adminManagePost()	The page of admin to manage
		post.
Algorithm	adminManagePost() START DISPLAY ContentOfPost FROM Post	

IF (click=Delete) at AdminManagePost interface
<b>DELETE FROM Post WHERE ContentOfPost = ContentOfPost</b>
END IF
END

 Table 2. 40: AdminManagePost controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for user to manage	
	complaint.	
Attributes	Attributes Name	Attributes Type
	Complaint_ID	INT
	ContentOfComplaint	String
	Status	String
	Reply	INT
Methods	Method Name	Description
	makeComplaint()	N/A
	<b>DeleteComplaint</b> ()	
Algorithm	makeComplaint()	
	START	

READ ContentOfComplaint
SET Status = "PENDING"
INSERT INTO "Complaint", store Complaint_ID,
ContentOfComplaint, Status, Reply
Redirect to UserPage interface.
END
DeleteComplaint ()
STADT
SIAKI
DISPLAY ContentOfComplaint
DISPLAY Status
DISPLAY Reply
FROM Complaint
IF (click=Delete) at ManageComplaint interface
DELETE FROM Complaint WHERE Complaint_ID=
Complaint_ID
END IF
END

 Table 2. 41: MakeComplaint controller of UMP Confession Application

Class Type	Controller Class	
Responsibility	This controller is to receive and process data for admin to manage complaint.	
Attributes	Attributes Name	Attributes Type
	Complaint_ID	INT
	ContentOfComplaint	String
	Status	String
	Reply	INT
Methods	Method Name	Description
	ReplyComplaint()	N/A
Algorithm	ReplyComplaint()	
	START	
	DISPLAY ContentOfComplaint	
	DISPLAY Status FROM Complaint	
	READ Reply	
	UPDATE INTO "Complaint" WHERE Complaint _ID = "Complaint _ID" Redirect to AdminPage interface.	
	END	

 Table 2. 42: ManageComplaint controller of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for user	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String
	НР	String
	image	IMAGE
	validornot	INT

Table 2. 43: User model of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for admin	
Attributes	Attributes Name	Attributes Type
	ID	INT
	username	String
	password	String

Table 2. 44: Admin model of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for post	
Attributes	Attributes Name	Attributes Type
	Post_ID	INT
	TypeOfPost	String
	ContentOfPost	String
	ImgofPost	IMAGE
	ID	INT

 Table 2. 45: Post model of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for comment	
Attributes	Attributes Name	Attributes Type
	Comment_ID	INT
	ContentOfComment	String
	Post_ID	INT

 Table 2. 46: Comment model of UMP Confession Application

Class Type	Entity Class		
Responsibility	A database table for order		
Attributes	Attributes Name     Attributes Type		
	Order_ID	INT	
	Order_Type_ID	INT	
	DetailsOfOrder	String	
	Comment	String	
	ID	INT	

Table 2. 47: Order model of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for announcement	
Attributes	Attributes Name	Attributes Type
	Announcement _ID	INT
	ContentOfAnnounce	String

 Table 2. 48 : Announcement model of UMP Confession Application

Class Type	Entity Class	
Responsibility	A database table for order type	
Attributes	Attributes Name	Attributes Type
	Order_Type_ID	INT
	OrderType	String

### Table 2. 49: Order type model of UMP Confession Application

Class Type	Entity Class		
Responsibility	A database table for Complaint		
Attributes	Attributes Name     Attributes Type		
	Complaint_ID	INT	
	ContentOfComplaint	String	
	Status	String	
	Reply	String	

 Table 2. 50: Complaint model of UMP Confession Application

Field Name	Description	Data Type	Constraint
ID	ID of user	INT	РК
username	Username	VARCHAR(20)	
password	Password for login	VARCHAR(20)	
HP	Phone number of the user	VARCHAR(12)	
Image	Image upload for admin to check to approve them as valid user	IMAGE	
validornot	To check wheatear the user is valid or not which 1 is valid and 0 is not valid	INT	

#### **DATA DICTIONARY** 2.2

### Table 2. 51: Data Dictionary for User

Field Name	Description	Data Type	Constraint
ID	ID of admin	INT	РК
username	Username	VARCHAR(20)	
password	Password for login	VARCHAR(20)	

### Table 2. 52: Data Dictionary for Admin

Field Name	Description	Data Type	Constraint
Post_ID	ID of post	INT	РК
TypeOfPost	Type of post that user make	VARCHAR(50)	
ContentOfPost	Content of post	VARCHAR(50)	
ImgofPost	Image upload for post	IMAGE	
ID	ID of the user who make the post	INT	FK

Table 2. 53: Data Dictionary for Post

SOFTWARE DESIGN DESCRIPTION (SDD) FKOM

Field Name	Description	Data Type	Constraint
Comment_ID	ID of comment	INT	РК
ContentOfComment	Content of comment	VARCHAR(50)	
Post_ID	ID of post that have this comment	INT	FK

 Table 2. 54: Data Dictionary for Comment

Field Name	Description	Data Type	Constraint
Announcement _ID	ID of announcement	INT	РК
ContentOfAnnounce	Content of announcement	VARCHAR(50)	

Table 2. 55: Data Dictionary for Announcement

Field Name	Description	Data Type	Constraint
Order_ID	ID of order	INT	РК
Order_Type_ID	Type of order that user choose	INT	FK
DetailsOfOrder	Details of the order that user want to make	VARCHAR(50)	
Comment	Comment that used to update status of the order by the user who responsible to this order type.	VARCHAR(50)	
ID	ID of the user who make the order	INT	FK

Table 2. 56:	Data Dictionary	for Order
--------------	-----------------	-----------

, 7

Field Name	Description	Data Type	Constraint
Order_Type_ID	ID of order type	INT	РК
OrderType	Type of order	VARCHAR(50)	

### Table 2. 57: Data Dictionary for Order type

Field Name	Description	Data Type	Constraint
Complaint_ID	ID of complaint	INT	РК
ContentOfComplaint	Content of complaint that user want to write.	VARCHAR(50)	
Status	Status of complaint which is PENDING when user make the complaint and RECEIVE when admin reply the complaint.	VARCHAR(50)	
Reply	Reply from admin on a specific complaint.	VARCHAR(50)	

 Table 2. 58: Data Dictionary for Complaint

# **APPENDIX C**

# UAT (User Acceptance Test)

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by:

Name: LIM KAH CHUN

Signature:

Date: 30/12/2023

UAT Form from author

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by: Name: ANG SUZANNE

M.

Signature: Date: 12/1/2023

UAT Form use for testing (User 1)
No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by: Name: KONG KEI

Signature: Date: 12/1/2023

UAT Form use for testing (User 2)

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by:

Name: LIM YI CI

Signature:

Date: 12/1/2023

UAT Form use for testing (User 3)

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by:

Name: TEO VOON CHUAN

fy.

Signature: Date: 14/1/2023

UAT Form use for testing (User 4)

No	Module	Activities	Status		Comments
			Yes	No	
1	Login	Register	/		
2		Login as admin	/		
3		Login as user	/		
4		Check account	/		
5	Manage post	Make post	/		
6		Delete post	/		
7	Add comment	Manage comment	/		
8	Manage order	Make order	/		
9		Edit order	/		
10		Delete order	/		
11		View order and give comment	/		
12	Manage	Make announcement	/		
	announcement				
13		Delete announcement	/		
14	Manage order	Add new order type	/		
	type				
15		Delete order type	/		
16	Manage	Make complaint	/		
	Complaint				
17		Reply complaint	/		

This test has been performed by: Name: HO SHU XIAN

( file

Signature:

Date: 19/1/2023

UAT Form use for testing (User 5)