

DEVELOPMENT OF A COMPUTERIZED INVENTORY  
MANAGEMENT SYSTEM (IMS) FOR  
INDUSTRY APPLICATION

OOI CHOON KHENG

B. ENG (HONS.) MECHATRONICS ENGINEERING  
UNIVERSITY MALAYSIA PAHANG





## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : OOI CHOON KHENG

Identification Card No : 911118-10-5885

Title : DEVELOPMENT OF A COMPUTERIZED  
INVENTORY MANAGEMENT SYSTEM (IMS)  
FOR INDUSTRY APPLICATION

Academic Session : 2014/2015

I declare that this thesis is classified as:

**CONFIDENTIAL**

(Contains confidential information under the  
Official Secret Act 1972)

**RESTRICTED**

(Contains restricted information as specified by  
the organization where research was done)\*

**OPEN ACCESS**

I agree that my thesis to be published as online  
open access (Full text)

I acknowledge that University Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of University Malaysia Pahang.
2. The Library of University Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Author's Signature)  
\_\_\_\_\_

\_\_\_\_\_  
(Supervisor's Signature)  
Name of Supervisor

Date: \_\_\_\_\_

Date: \_\_\_\_\_

DEVELOPMENT OF A COMPUTERIZED INVENTORY  
MANAGEMENT SYSTEM (IMS) FOR  
INDUSTRY APPLICATION

OOI CHOON KHENG

Thesis submitted in partial fulfilment of the requirements for the award of  
the degree of B. Eng (Hons.) Mechatronics Engineering

Faculty of manufacturing engineering  
UNIVERSITY MALAYSIA PAHANG

JUNE 2015

## EXAMINER'S DECLARATION

I certify that the project entitled "Development of a Computerized Inventory Management System (IMS) Software for industry application" is written by Ooi Choon Kheng. I have examined the final copy of this thesis and in my opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Mechatronics Engineering. I am here with recommend that it be accepted in partial fulfillment of the requirements for the degree of B. Eng (Hons) Mechatronics Engineering.

Signature :

Name :

Date :

### **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Mechatronics engineering.

Signature :

Name of Supervisor :

Position :

Date :

## STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :

Name :

ID Number :

Date :



## ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude towards my project supervisor, Dr Nizam Uddin Ahamed for his support and guidance that giving me the strength to accomplish this final year project. Besides that, I would like to thanks him for giving me the opportunity to work on this project under his supervisor. He has provided me with great approaching and feedback on every step of the way and as a result of that, I have learned and grown well.

My appreciation also goes to my parents, sibling, lecturers and friends for their understanding and support throughout the whole duration of completing this project. Thanks to their love, encouragement, and mental support that they has given me, I have been able to complete the project.

Last but not least, I would also like to acknowledge the following websites. Such as, codeproject, github, Microsoft, zxing, Aforge and other website for the free source library they provided and also the knowledge they share that help me to complete the software of the project.

## ABSTRACT

Inventory Management System (IMS) are widely used in industry nowadays to increase the efficiency of item flow. The traditional method of IMS involved a lot of paper work such as spreadsheet and order list which will be more difficult to manage as the storage grew bigger. Therefore, inventory management system can be computerized to further increase the efficiency of IMS. This project aim to develop IMS software can store a large amount item data. Moreover, it has check in/out function and search function that run using scan QR Code. Besides that, the software also has the feature of low stock warning function to warn the user if the item has low quantity. To develop the software, first the criteria of the software are determined followed by choosing the software development tools to develop the software that can achieve the criteria of the software. Next the software will be developed and debugged. This paper focuses on developing IMS software for SMEs and biomedical field company usage.

## ABSTRAK

Sistem Pengurusan Inventori (IMS) digunakan secara meluas dalam industri pada masa kini untuk meningkatkan kecekapan aliran item. Kaedah tradisional IMS melibatkan banyak kertas kerja yang akan menjadi lebih sukar untuk diuruskan sekiranya penyimpanan barang berkembang semakin besar. Oleh itu, sistem pengurusan inventori boleh berkomputer untuk meningkatkan lagi kecekapan IMS. Projek ini bertujuan untuk membangunkan perisian IMS yang boleh menyimpan item data yang berjumlah besar. Selain itu, ia mempunyai fungsi daftar masuk / keluar dan fungsi pencarian yang melibatkan menggunakan imbasan QR Kod. Selain itu, perisian ini juga mempunyai ciri fungsi amaran kiraan barangan rendah untuk memberi amaran kepada pengguna jika ada item yang mempunyai kuantiti yang rendah. Untuk membangunkan perisian IMS, pertama sekali, kriteria perisian akan ditentukan mengikut pemilih alat-alat pembangunan perisian untuk membangunkan perisian yang boleh mencapai kriteria perisian. Seterusnya perisian IMS akan dibangunkan dan debugged. Thesis ini memberi tumpuan kepada membangunkan perisian IMS untuk penggunaan SMEs dan bioperubatan syarikat.

## TABLE OF CONTENTS

		<b>Page</b>
<b>EXAMINER’S DECLARATION</b>		ii
<b>SUPERVISOR’S DECLARATION</b>		iii
<b>STUDENT’S DECLARATION</b>		iv
<b>ACKNOWLEDGEMENT</b>		v
<b>ABSTRACT</b>		vi
<b>ABSTRAK</b>		vii
<b>TABLE OF CONTENTS</b>		viii
<b>LIST OF TABLES</b>		xi
<b>LIST OF FIGURES</b>		xii
<b>LIST OF ABBREVIATIONS</b>		xiv
<b>CHAPTER 1</b>	<b>INTRODUCTION</b>	
1.1	Introduction	1
1.2	Background	1
1.3	Problem Statement	3
1.4	Objectives	3
1.5	Scope of Research	3
<b>CHAPTER 2</b>	<b>LITERATURE REVIEW</b>	
2.1	Introduction	4
2.2	Method of Literature Search	4
2.3	Gap finding	5
	2.3.1 Literature Search Results	5
	2.3.2 The Technology Used	10
	2.3.3 The Software Development Tool Used	10
	2.3.4 The Software Features of the Previous Projects	11
2.4	Conclusion	11

### **CHAPTER 3            METHODOLOGY**

3.1	Introduction	12
3.2	Method to Develop IMS Software	12
3.3	Determine the Condition for the Software	13
3.4	Determine the Software Features	14
3.5	Determine the Software Development Tools to be Used	14
3.5.1	Coding Library Comparison	16
3.5.1.1	QR Code Library	16
3.5.1.2	Connect to MySQL (XAMPP) Library	20
3.5.1.3	Create GUI Library	20
3.5.2	Choice of Software Development Tools	20
3.6	Conceptual Software Flow Chart	21
3.7	Conceptual Software GUI	25
3.8	Design the Database	28
3.9	Debug Software	28
3.10	Conduct Survey	28

### **CHAPTER 4            RESULTS AND DISCUSSION**

4.1	Introduction	29
4.2	Overview of IMS Software	29
4.3	Developed Database	30
4.4	Developed GUI	31
4.5	Survey Conducted	38

### **CHAPTER 5            CONCLUSION AND RECOMMENDATION**

5.1	Introduction	39
5.2	Conclusion	39
5.3	Recommendation	40

<b>REFERENCES</b>	41
<b>APPENDICES</b>	
A        IMS Software Coding	43
B        Example of Survey Form	59
C1       Final Year Project 1 Gantt Chart	60
C2       Final Year Project 2 Gantt Chart	61

**LIST OF TABLES**

<b>Table no.</b>	<b>Title</b>	<b>Page</b>
2.1	Summary studies comparing the methods and technology used to develop an Inventory Management System	9
3.1	List of current IMS software available in market	15
3.2	List of supported Module by ZXing library	17
3.3	List of Code supported by ZXing library	18
3.4	List of Code supported by BARCODE IMAGE GENERATION library	19
3.5	Summary of coding library	19
4.1	Survey result	38

## LIST OF FIGURES

<b>Figure no.</b>	<b>Title</b>	<b>Page</b>
2.1	Flow Chart of method used for literature search	5
3.1	Method to develop Inventory Management System (IMS) Software	13
3.2	Software Flow Chart Part 1	21
3.3	Software Flow Chart Part 2	22
3.4	Software Flow Chart Part 3	23
3.5	Software Flow Chart Part 4	24
3.6	Software Flow Chart Legends	25
3.7	Login Tab Conceptual Design of GUI	26
3.8	Itemview Tab Conceptual Design of GUI	26
3.9	Add item Tab Conceptual Design of GUI	27
3.10	QR code identification method	27
4.1	Overview of IMS	29
4.2	Overview of the database	30
4.3	Login_data data table example	30
4.4	ims_test data table example	31
4.5	000002 data table example	31
4.6	User selection form	32
4.7	Register form	32
4.8	Login form	33
4.9	Mainpage form	34
4.10	Notification form	34
4.11	Item box form	35
4.12	Add item form	35



4.13	Photo Taker form	36
4.14	QR Code scanner form	36
4.15	Check out form	37

## LIST OF ABBREVIATIONS

IMS	Inventory Management System
GUI	Graphical User Interface
QR Code	Quick Response Code

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

This chapter is written on the background of Inventory Management System (IMS), problem statement, objectives to be achieved and scope of research of “Development of a computerized Inventory Management System (IMS) software for industrial application”.

#### **1.2 BACKGROUND**

To keep business running at a good pace, a good management is required to constantly maintains a good balanced between the items keep in storage and the items going in and out. In order to achieve a good management, inventory management system as the tool to monitor the inventory level and items status is used.

Before the era of technology, Inventory Management System was a system involving paperwork to record down all the item status. It was widely used by companies to manage their storage. However as the company grow bigger and the storage increase in size where sometimes the storage has to be divided to a few location, thus the Inventory Management System would use up a lot of spaces and money to keep track of the item as it used more paper to keep track of the item and at the same time the papers need more spaces to stored. It will become less efficiency as longer time is needed to arrange the information from all the storage and find the information about a certain item within that huge pile of papers.

To overcome this, computerized Inventory Management System has been introduced to increase the efficiency despite having a large storage and making possible to search the information or a specific item in a short time. Nowadays, the computerized Inventory System has been further improved by the technology we have today and even real time item monitoring is possible for the user can make modify or view the storage status and item status with a few clicks by their fingertips.

Computerized Inventory Management System involved a computer loaded with a software capable of interacting with user using Graphical User Interface and capable of registering new items, deleting items, modifying items details, generate a label for each item, categorizing each item accordingly and store the data in a database and display it in the Graphical User Interface for the user to view or modify it. The labels are later on being used to label the items and a scanner is used to scan the labels to identify the item and view its detail on the Graphical User Interface.

There are many type of methods that can be used to label and identify an item. One of the oldest methods is barcode system. By generating unique barcode through the computer after registering a new item and its details and label the item with the generated barcode, user will be able to identify all the status of the specific item and search the item more efficiently at the computer by simply scanning the barcode using a barcode scanner. Besides that, QR code system is also one of the favourite systems used to identify the item status and it works similiarly to the barcode system which require user to input the data to generate a code unique to a specific item and scan the code to obtain the item information.

This project aims to develop and validate an user-friendly Inventory Management System (IMS) software capable of storing large amount of data on items for usage in biomedical field, Small-Medium-Entrepreneurs companies and any other organization or company with a low cost.

### **1.3 PROBLEM STATEMENT**

Inventory management has been practiced by many companies to increase efficiency of item flow. Nowadays, Small and Medium Entrepreneurs (SMEs) still practice inventory management using traditional method such as spreadsheet and order list although they are in the era of technology. Hence, Inventory Management System can be computerized to further increase the efficiency of item flow and easier to manage.

### **1.4 OBJECTIVES**

- I. To identify the most suitable software development tool to design Inventory Management System (IMS).
- II. To design and develop a low cost Inventory Management System (IMS) software.
- III. To validate and verify the Inventory Management System (IMS) software.

### **1.5 SCOPE OF RESEARCH**

- I. The software will be used in field of biomedical to organize the storage.
- II. The software will be used by SMEs companies to organize their storage.
- III. The software can be used by University Malaysia Pahang to organize university property.
- IV. The software can be used by the laboratory in Faculty of Manufacturing Engineering, University Malaysia Pahang.

## **CHAPTER 2**

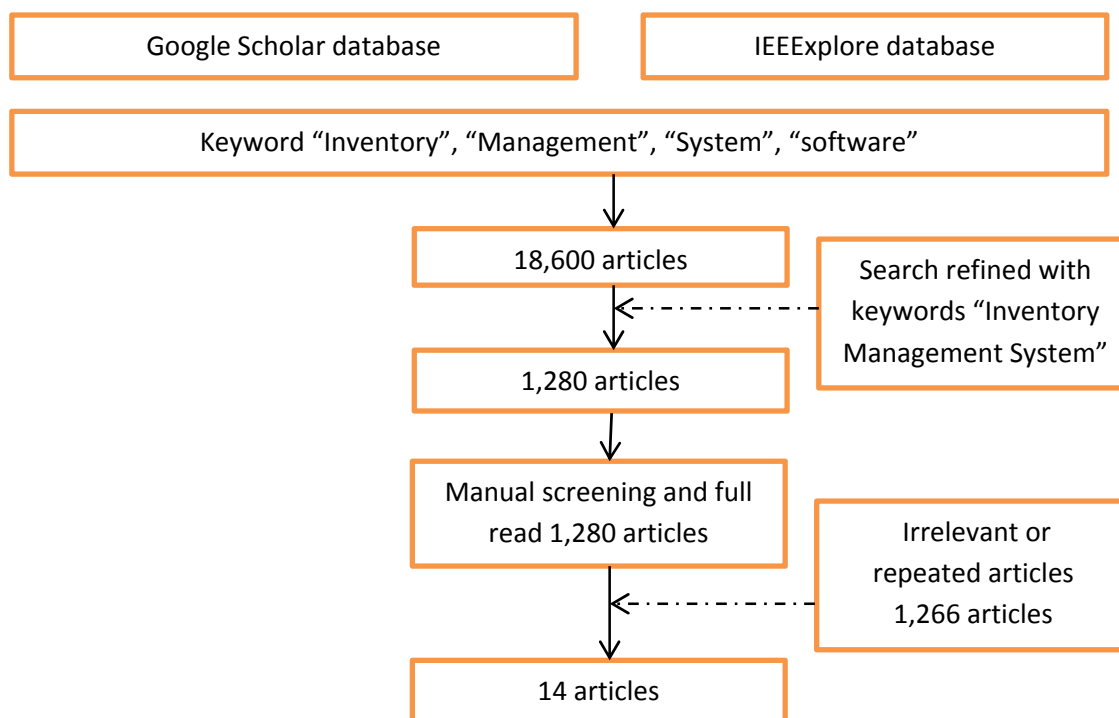
### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

In this section, the researches and studies done on article with previous similar projects which is to develop an Inventory Management System (IMS) software will be discussed. In addition, method of doing the literature research will also be discussed. The studies done are mainly to determine the software development tool used and technology used to create the IMS software in the previous projects and also the features of the IMS software. Then the knowledge gained from this section will be used as references to determine the direction of this project to create an unique Inventory Management System software.

#### **2.2 METHOD OF LITERATURE SEARCH**

By using the keyword “Inventory Management System Software”, a search on the literature is conducted on the studies published between year 2005 and 2014, in the Google Scholar database and IEEExplore database. Then the search was refined by adding keywords “Inventory Management System”. Then the result is further filtered using the criteria. The criteria included were: (1) Development of Inventory Management System and (2) Designing and analysis of Inventory Management System. The criterion excluded was: (1) Simulation of Inventory Management System.



**Figure 2.1:** Flow Chart of method used for literature search

## 2.3 GAP FINDING

This part will focus on discussing the literature search results and the literatures found. The literatures will be further analyzed to identify the technology and software development tool used to develop the Inventory Management System software in the previous projects and determine the gap of the previous projects. Then, the technology and software development tool of this project and features of this software will be determine based on the summary make from the literature analysis.

### 2.3.1 LITERATURE SEARCH RESULTS

The study shows that a total of 16800 articles that fitted the criteria words: "Inventory", "Management", "System", and "Software" were retrieved from the online database used. A refined search using the keyword "Inventory Management System" was made and retrieved 1280 articles. Then 1266 articles were excluded according to the criteria set, and only 14 articles were being analyzed further. The method is as shown in figure 2.1. In the table 2.1, the studies are done using Google scholar and

IEEEExplore. From number 1 to 12, the papers are found in Google Scholar with the exact keyword “Inventory Management System” and for number 13 to 14, the papers are found in IEEEExplore with the exact keyword “Inventory Management System”.



No.	Years	Title	Software Used	Technology used	Objective	Methodology	Focus	Author
1	2014	AN RFID based supply chain Inventory Management Solution for the petroleum development industry: A case study for Shell Nigeria [1]	Not mentioned	RFID	To show that an RFID supply chain inventory management system can be a viable solution to the infinite challenges faced in the operation of the storage in the petroleum development industry	Design solution architecture with management application as the center connected to inventory database and to mobile RFID reader within store and Fixed RFID reader at store exit > the tag readers will be linked to ERP system database to detect and trace the items movements	enhance the operational efficiency of the logistic and warehouse organization	Inalegwu Adoga and Raul Valverde
2	2014	Development of Mobile Enterprise Inventory Management System Application with CBD [2]	Android software, AppGyver, HTML5, SQL server	Barcode/QR code enabled smartphone	To design a mobile application intended for enterprise used, run in mobile devices with the CBD software engineering technique	Identify the components of the system > Break down functions > develop the system using SDLC methodology (planning > analysis > design > implementation)	high security data server, mobility (smart phone apps used in real time), increase efficiency, accuracy, software stability, understandability	Yvette E. Gebogo and Haeng-Kon Kim
3	2013	Application of RFID on equipment parts readiness management system of semiconductor packaging plant [3]	Visual Basic 6.0	RFID tags	To apply RFID on equipment parts readiness management system of semiconductor to decrease production time	Design system hardware based on operation procedure of REPE-MS > Design the RFID operation procedure > Design the system > Design the software > Compare the performance of the gadgets now and before	Enhance gadgets performance and efficiency	Wei-Ling Wang, Chia-Tzu Huang, Shi-Jen Wang and Chia-Pao Chang
4	2011	An Intelligent Warehouse Stock Management and Tracking System based on Silicon Identification Technology and 1-wire Network Communication [4]	Microsoft .Net, TIMEX C API, and Microsoft MySQL	Silicon Identification Technology (One-wire Network Communication)	To develop a wide variety Intelligent Warehouse Stock Management and Tracking system by introducing iButtons run on one-wire network protocol for various application	1-wire communication protocol and iButtons enhance with chain function communication concept > Conditional Read ROM function is added > Design GUI	Low cost, iButtons has high reliability and durability, Technological Acceptability, Rapid Development Time	Kai-Xin Tee
5	2011	The Application of Wi-Fi RTLS in Automatic Warehouse Management System [5]	Not mentioned	PIC16C73, RFID tags, bar code, wireless modules	To solve the management of warehouse related information by applying Wi-Fi RTLS technology	Design system hardware using MCU PIC16C73 as controller > system software design based on system functions	Increase item transfer capacity within one day, automatic item identification and item data tracing, verify and supervision function, cost reduction	Xiangguo Ma
6	2012	Design of an RFID-based Inventory Control and Management System: A case study [6]	Not mentioned	RFID	To enhance product life cycle management by collecting accurate real-time transactions of physical stock items data.	Identify the current management problem > Identify the work flow in and out > Design the system > Assess the workplace compatibility with RFID > Implement system >	Enhance the system's capability of identifying transactions and locations of tagged stock items reduce cost and increase efficiency	Jacky S.L. Ting and Albert H.C. Tsang

7	2011	The use of RFID and web 2.0 Technologies to Improve Inventory Management in South African Enterprises [7]	Web 2.0 tool	RFID	To introduce the use of a Web 2.0 tool such as twitter fuse with RFID technology to update the inventory manager on the inventory status anytime and anywhere.	Design and propose architecture for inventory management where internet is the center of the control which connect the local server to a remote server, QNS server and to user cellphone for user to access	Increase efficiency, mobility and real time update	Sizakele Mathaba, Nomusa Dlodlo, Andrew Smith and Mathew Adigun
8	2013	Wireless Infrastructure for Oil and Gas Inventory [8]	XML	RFID, Wireless Sensor Network	To develop an online inventory management system which enables monitoring the essential events in oil and gas offshore environment using a user intuitive web interface application	Design the system includes 5 modules (RFID system, weight measurement system, wirelessHART compliant node for communication, WirelessHART compliant gateway for network management and co-ordination and Toile Inventory Web server application enable remote access)	Increase efficiency, long range management	Shanthi Yellingri, Apala Ray, Maalikarjun Kande
9	2010	Intelligent service-integrated platform based on the RFID technology and software agent system [9]	Not mentioned	RFID, Software Agent	To design an intelligence platform with framework based on software agents to build an integrated information system mechanism integrated with RFID technology act as the trigger point for commodity message receiver as smart shelf to enhance performance of the sales outlets, customer service and identify popular commodity	Design the system architecture > determine the types of software agents to be used and its function > Determine the type of RFID to be used and its function > Design the architecture of the intelligent service-integrated platform	real time customers demand and product status control, increase accuracy of forecast of inventory data, decrease issues on shortage and replenishment, enhance product and customer support	Kun-Chieh Yeh, Ruey-Shan Chen and Chia-Chen Chen
10	2010	The Design and Implementation of the Integrated Supply Chain Management System Based on UML and J2EE Technology [10]	J2EE	UML, J2EE	To plan, build and apply Integrated Supply Chain Management System (SCMS) based on J2EE and UML	Plan the Architecture of the system based on J2EE technology > Design and apply integrated supplychain management system	optimize the supply chain performance	Yongchun Wu, Zhenjian Jiang, Na Han
11	2009	A RFID-based Material Supply Management System in Automatic Vehicle Assembly Streamline [11]	MySQL server, Sun Java System RFID, J2EE	RFID, J2EE	To build a RFID-based item source management in automatic vehicle assembly streamline for control over the material consumption of each work space in real time	Plan the system architecture > prototyping > Implement the System	enhance efficiency and accuracy of item supplement in the workshop	Wexing Su, Kunyuan Hu, Lei Zhang, Lianbo Ma
12	2008	Development and Implementation of Radio-frequency Identification (RFID) technology for inventory management system: A case study [12]	Visual Basic 6.0, Microsoft Access data base packages	RFID	To develop and implement RFID technology for supply store inventory management system	Analysis the data of previous system > Determine the flow of data > Design and develop system hardware > Design and develop system software	increase efficiency and effectiveness of the system	Wan Harun Wan Hamid and Loh Chee Hong

13	2013	Research and design of the intelligent inventory management system based on RFID [13]	SQL server 2005, web server	GSM, RFID	To analyses the data flow of intelligent inventory system from the perspective of business and function and design and develop a specific framework program and function modules of intelligent inventory management system based on IOT RFID technology.	Design network structure > Design system function and structure > Design system database	Saving logistics costs, reducing labors costs, improving operation accuracy and storage efficiency	Xiaojun, Jing and Peng Tang
14	2011	The Design and Implementation of Distributed Inventory Management System Based on the Intranet Architecture [14]	SQL server 2005, Web server, ASP.NET + C# language	Non	To investigates and plans the distributed inventory management system function model, and propose system design and implementation methods, and implement the valid management and retrieve distributed inventory information fast and accurately	Functional model based on analysis on all of the products status is build > Develop the system software	increase information efficiency, reduce cost and time consumption, increase production varieties, increase income, assist in decision making, enhance communication platform for optimal material reserve	Liling Xia

**Table 2.1:** Summary studies comparing the methods and technology used to develop an Inventory Management System

### 2.3.2 THE TECHNOLOGY USED

Most of the studies (Wan Hanun Wan Hamid and Loh Chee Hong (2008); Xiaojun Jing and Peng Tang(2013); Weixing Su et al.(2009); Kun-Chieh Yeh et al.(2010); Sharthi Vellingiri et al(2011); Sizakele Mathaba et al.(2011); Jacky S.L. Ting and Albert H.C. Tsang(2011); Wei-Ling Wang et al. (2013); Inalgewu Adoga and Raul Vaverde (2014)) used RFID as the medium to identify and track the items in the inventory[1,3,4,6,7,8,9,11,12,13]. Whereas the studies by Kai-Xin Tee (2011) used Silicon Identification Technology to identify and track the items [4] and Yvette E. Gelogo and HaengKon Kim (2013) used Barcode and QR-code to identify and track the item [2]. Besides that, Xiangguo Ma (2011) has used combine technology of RFID and barcode to identify and track the item [6].

### 2.3.3 THE SOFTWARE DEVELOPMENT TOOL USED

For developing the Graphical User Interface (GUI), Wan Hanun Wan Hamid and Loh Chee Hong (2008) and Wei-Ling Wang et al.(2013) used Visual Basic 6.0 [3,12]. As for Xiaojun Jing and Peng Tang(2013); Liling Xia (2011); Sizakele Mathaba et al.(2011); Kai-Xin Tee (2011) they used Microsoft .NET Framework to create the Graphical User Interface [7,13,14]. Weixing Su et al.(2009); Yongchun We, Zhenjian Jiang, Na Han(2010) used J2EE to create the Graphical User Interface [10,11]. Yvette E. Gelogo and HaengKon Kim (2013) used Android software development tools, AppGvyer to create the GUI for mobile phone and HTML5 for computer [2]. On the other hand, Sharthi Vellingiri et al.(2011) used XML to develop GUI [8].

As for database server, most of the studies (Yvette E. Gelogo and HaengKon Kim (2013); Kai-Xin Tee (2011); Weixing Su et al.(2009); Xiaojun Jing and Peng Tang(2013); Liling Xia (2011) ) used SQL server to build the database [2,3,4,11,14]. Wan Hanun Wan Hamid and Loh Chee Hong (2008) used Microsoft Access to create the database[12].

### 2.3.4 THE SOFTWARE FEATURES OF THE PREVIOUS PROJETS

One of the aims of all of the articles except Kai-Xin Tee (2011) is to improve the efficiency of inventory management [1,2,3,5,6,7,8,9,10,11,12,13,14]. Liling Xia (2011); Xiaojun Jing and Peng Tan (2013); Kai-Xin Tee (2011) have additional aim to reduce the costs for inventory management system [4,11,13]. Moreover, Yvette E. Gelogo and Haeng-Kon Kim (2013) has others aims that are increase mobility, software accuracy, stability and understandability and high security data server [2]. While Kai-Xin Tee (2011) mainly focus on introducing new technology “iButtons” and its acceptability with its rapid development time and high reliability and durability features [4]. In addition, Sizakele Mathaba et al.(2011) has aim to provide real time update besides its focus to increase efficiency and mobility [7]. Kun-Chieh Yeh et al.(2010) has aim to provide real time customer demand and merchandise status control to improve product service and customer supports [9].

## 2.4 CONCLUSION

Based on the studies done on all 14 articles, it is discovered that the usage of software development tool such as Visual Studio, Code Block and Dev C++ and android software development tools such as Eclipses and Unity have not been tried. GUI which enable the user to access the database from the main server and also from mobile phone. Most of the projects adapt Microsoft .NET to make GUI. The database is made using SQL server and only one project used Access to create the database. There is no attempt on using excel to create a database or create a cloud database. As for the technology, RFID is used in most of the projects to identify and track the item and barcode/QR-code is rarely used. As for focus of the projects, overall aim to improve efficiency of the Inventory Management System. There are only 2 articles paid extra attentions on providing real time control or monitoring. In addition, only 1 article paid more attention on increase mobility, software accuracy, stability and understandability and high security data server.

As a conclusion, the features of this software will be to increase efficiency, real time control and monitoring, increase understandability and low stock alarm feature.

## **CHAPTER 3**

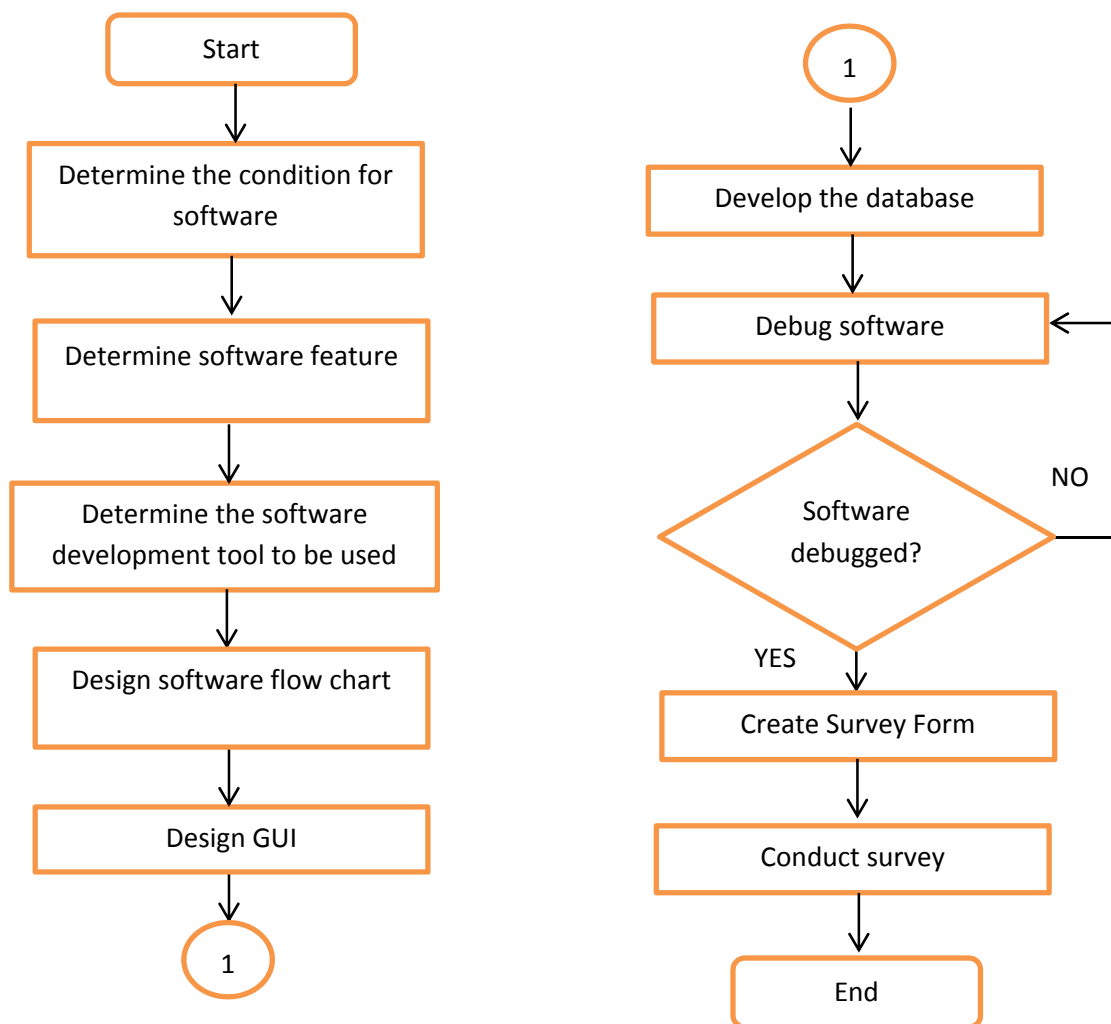
### **METHODOLOGY**

#### **3.1 INTRODUCTION**

In this section, the method and software development tool used to develop the Inventory Management System (IMS) will be discussed. The method used to develop the software will be explained in details together with figures and flow charts.

#### **3.2 METHOD TO DEVELOP IMS SOFTWARE**

First, a condition will be set on the software based on the usage of the software. Then, the features of the software are determined based on the needs. Next, a data flow diagram is created to determine the flow of data from server to user. The software development tools is chosen based on the limitation and features. Then, the software flow chart is created based on the limitation and features. Next, the conceptual design of the GUI is created to visualize how the software will looks like when it is done. Next the GUI will be built with the full function including the QR Code Encoder and Decoder follow by the creation of database. The software will be debugged until it can run smoothly. Lastly, a survey form is created to gather data on user satisfaction and the software will be tested by 10 persons and then fill up the survey form.



**Figure 3.1:** Method to develop Inventory Management System (IMS) Software

### 3.3 DETERMINE THE CONDITION FOR SOFTWARE

A condition is set to act as the boundary of the software so that it will only function within the boundary based on the usage of the software. According to the objective of this project, the software will be used within companies of SMEs and biomedical field. Hence, it is assume that the software will be used locally, within the company and there is only one storage area since the company is not too large. Next the software will be made available only on the server computer to register the check in/ out of the item and check the status of items to prevent multiple access at the same time which will causes data to overload.

### **3.4 DETERMINE SOFTWARE FEATURE**

The features of the software are determined based on the expected outcome and the result of the research done in chapter 2 literature review. First, the software have to be user friendly so that it is easily understand and can be used by the user without the needs of training. Next, the basic feature of the software is to register item for both check in and check out. Besides that, user can add in description to help user to recognize the item. The database will be updated in real time. In addition, users will able to scan the QR code on the item to determine and extract the item information and status at any time from the server computer. The database is MySQL database that will be hosted by a free software XAMPP. Moreover, there is a low stock reminder function to remind the user that the item quantity of the certain item is lower than the 5 so that the user can restock accordingly.

### **3.5 DETERMINE THE SOFTWARE DEVELOPMENT TOOLS TO BE USED**

A research have been done on the current IMS software in the market, and the following table 3.1 is a list of IMS software available in current market. Based on table 3.1, it can be concluded that the language used to develop the software is normally separated into two group C programming(C, C++, C#) and JAVA programming. As listed in table 3.1, most of the software is windows based which means that they are developed using C, C++, or C# language that runs on windows platform. There are a few of the software is developed using JAVA which can be website based or android based. JavaScript can be added in PHP to run on website and JAVA language can be used to develop Android apps or even to make windows based apps.



Name	Reference site	System requirement	Features	Hardware supports	Usage	Customization
Windward software System Five Inventory Management	<a href="http://wvs5.com/index.php?content=inventory_control.txt">http://wvs5.com/index.php?content=inventory_control.txt</a>	Desktop: Microsoft Windows and Server: microsoft windows server, linux or Unix Novell Netware (* Insist to be Novell Certified)	Inventory Reporting, Sales History, Price Labels, Alternate parts, Sales history, Manage Return Merchandise Authorization, Back order, Special order	Receipt printers, dot matrix printer, laser printer, Barcode hardware(CCD, Laser, Linear imagers), magnetic card reader	Small, medium warehouse	Yes
Windward software System Five Advanced Inventory Management	<a href="http://wvs5.com/index.php?content=advanced_inventory_control.txt">http://wvs5.com/index.php?content=advanced_inventory_control.txt</a>	Desktop: Microsoft Windows and Server: microsoft windows server, linux or Unix Novell Netware (* Insist to be Novell Certified)	Create location, Stock movement, Stock maintenance, auditing stock, Inventory Reporting, Sales History, Price Labels, Alternate parts, Sales history, Manage Return Merchandise Authorization, Back order, Special order	Receipt printers, dot matrix printer, laser printer, Barcode hardware(CCD, Laser, Linear imagers), magnetic card reader	Big, multiple warehouse	Yes
Visco Software	<a href="http://viscosoftware.com/secondary/product/requirements.htm">http://viscosoftware.com/secondary/product/requirements.htm</a>	Microsoft.NET (web based software), IE 6.0 or later, MS office 2003 or later, high speed internet connection	Sales orders & purchase order maintenance, logistic management, inventory adjustment, sales tools, financial tools, customer tracking, system management, unit and currency converter, product/customer/vendor details	printer, barcode by mobile device	Small, medium warehouse	Yes
Waspbarcode: inventory Management	<a href="http://www.waspbarcode.com/inventory-control/inventory-control-complete-solution#tab-1">http://www.waspbarcode.com/inventory-control/inventory-control-complete-solution#tab-1</a>	Windows xp sp3 or later, windows server 2003 or later, windows mobile 2003 or later, windows CE 4.2 or higher	Inventory tracking, item check in and out, barcode making, alerts, inventory review, audit and cycle count	Barcode printer, hardware	Small, medium, big, multiple warehouse	No
Handifox	<a href="http://www.handifox.com/">http://www.handifox.com/</a>	Windows xp sp3 or later, windows server 2003 or later, Quickboos software (2008 or later), PDA: MS windows mobile 5.0 or later, Android 2.3 or later	make barcode, create, edit and receive purchase orders, Count inventory and review in QuickBooks, build inventory assemblies, track inventory at multiple sites, access inventory data from quickbooks, make changes using QuickBook	Label printer, receipt printers	Small, medium, big, multiple warehouse	Yes
SCOUT: Easy Inventory Management	<a href="http://www.scoutsoft.com/easy-inventory-management/">http://www.scoutsoft.com/easy-inventory-management/</a>	Cloud-based, any mobile device	Alert system, planning, inventory review and details, cycle count, daily shipping and receiving, accounting, customer relationship management	printer, barcode by mobile device	Small, medium, big, multiple warehouse	Yes
Redbeam: Inventory Management	<a href="http://www.redbeam.com/inventory-management.html">http://www.redbeam.com/inventory-management.html</a>	Windows Xp or later, MS SQL 2000 or higher, MS office Excel	Detailed, vendor purchasing records, search function, serial, lot nad expiration date control, barcode system, inventory details	Barcode hardware, label or barcode printer	Small, medium warehouse	Yes
ASAP SYSTEMS: PASSPORT	<a href="http://www.asapsystems.com/inventory_control.php">http://www.asapsystems.com/inventory_control.php</a>	Any mobile phone, Windows xp pro or later	Inventory details, warehouse location detail and edit, First In First Out(FIFO) Control, export data to any file format, stock list, BOM generation, Supplier details	Barcode hardware	Small, medium big and multiple warehouse	Yes


**Table 3.1:** List of current IMS software available in market

### **3.5.1 CODING LIBRARY COMPARISON**

A research was done on the current free, open source library for coding that is related to developing this project based on the condition and features of software. The following is a list of code library for both C and JAVA with the details of each library that was found online.

#### **3.5.1.1 QR CODE LIBRARY**

ZXing ("zebra crossing") is an open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. It currently contains and supports the following module in table 3.2 and the following code format in table 3.3.

Module	Description
core	The core image decoding library, and test code
javase	JavaSE-specific client code
android	Android client Barcode Scanner 
androidtest	Android test app, ZXing Test
android-integration	Supports integration with Barcode Scanner via Intent
android-core	Android-related code shared among android, androidtest, glass
glass	Simple Google Glass application
zxingorg	The source behind zxing.org
zxing.appspot.com	The source behind web-based barcode generator at zxing.appspot.com

**Table 3.2:** List of supported Module by ZXing library

Source: Owen, S. (2015)

<b>ID product</b>	<b>ID industrial</b>	<b>2D</b>
UPC-A	CODE 39	QR CODE
UPC-E	CODE 93	DATA MATRIX
EAN-8	CODE 128	AZTEC (BETA)
EAN-13	CODABAR	PDF 417(BETA)
	ITF	
	RSS-14	

**Table 3.3:** List of Code supported by ZXing library

Source: Owen, S. (2015)

As for Open Source QR Code Library, it currently supports decoder for Java (J2SE, J2ME MIDP2.0/CLDC1.0), error correction by Reed-Solomon code and BCH code, pre-process camera-captured image for better performance such as filtering image noise and recognize angle of image automatically. Besides that, it also provides some sample applications. For example, CUI, GUI, MIDlet (for cellphones which support Java) and real time QR Code reader by Java Media Framework (JMF) and PC camera

QRCode Library 1.3.0 is a .NET component that can be used to encode and decode QRCode. QRCode Library provides functions to do encode content into a QR Code image which can saved in JPEG, GIF, PNG, or BITMAP formats and decode a QR Code image. The library can be used in any .NET 3.5 Windows Application, ASP.NET web application or Windows Mobile device application.

CODEBUDE/QRCoder is a simple library, written in C#.NET, which enables you to create QR Codes. It's licensed under the MIT-license. As for kenglxn/QRGen, it is a simple QRCode generation api for java built on top of ZXING library.

Barcode Image Generation Library can be used in c# and .NET and it supports the following format:

Code 128	Code11	Code 39 (Extended / Full ASCII)
Code 93	EAN-8	EAN-13
UPC-A	UPC-E	JAN-13
MSI	ISBN	Standard 2 of 5
Interleaved 2 of 5	PostNet	UPC Supplemental 2
UPC Supplemental 5	Codabar	ITF-14
Telepen	Pharmacode	FIM (Facing Identification Mark)

**Table 3.4:** List of Code supported by BARCODE IMAGE GENERATION library

Source: Barnhill, B. (2015)

Library Name	Language supported	Function
ZXing	C#, C++, .NET, JAVA	-Generate QRCode -Decode QRCode
OPEN SOURCE QR CODE LIBRARY	JAVA	-Decode QRCode
QR CODE LIBRARY 1.3.0	C#, .NET	-Generate QRCode -Decode QRCode
BARCODE IMAGE GENERATION LIBRARY	C#, .NET	-Generate QRCode
CODEBUDE/QRCODER	C#, .NET	-Generate QRCode
kenglxn/ QRGen	JAVA	- Generate QRCode

**Table 3.5:** Summary of coding library

### **3.5.1.2 CONNECT TO MYSQL (XAMPP) LIBRARY**

To connect to a MySQL database, it is required to download the suitable library from the MySQL webpage. For windows form, ODBC connector. As for the JAVA, J connector is required. For C# coding, the following website, provide the tutorial on how to connect the c# project with the MySQL. For java coding, the following website, provide the tutorial on how to connect the java project with the MySQL

### **3.5.1.3 CREATE GUI LIBRARY**

In Visual Studio 2013, the library used to create GUI is called System.Windows.Forms, it contains a lot of classes. Such as, control form, controls, layout, menus and toolbars, data and data binding, components and common dialog boxes for user to create a graphical user interface.(GUI) In Eclipse JAVA, the WindowBuilder is a plug in which enable the user to create GUI as easy as drag and drop the interface.

## **3.5.2 CHOICE OF SOFTWARE DEVELOPMENT TOOLS**

Based on the table 3.1 and information above, ZXing is the best library to be used for detecting and generating QR Code as it can supports more code format and multiple platform compare to the other choices of library which can be later be used to upgrade the software function of detecting more different kind of code. Besides that, it also provided a lot of example for references.

As for the connector to link to MySQL database, according to the information on the websites. Although the connector is different for different coding language, the method used to connect the database is the almost same and only the syntax is different. Based on the information above, both GUI library has the same function of enable the user to create GUI with just drag and drop the desired tools or interface into the form.

As a conclusion, based on the information in the whole section 3.4 this project will be using the technology QR Code to identify and track the item. Besides that,

Visual Studio will be used to create a GUI for computer interaction with the user. MySQL server will be used to create the database server and XAMPP will be used to host the database server. Then, the software condition and features can be achieved by using either C# or JAVA programming language. However, due to time constrain on this project and developer preference, this project will be using C# language to develop the software. Moreover, C# language is the widely used language to develop windows based software and from the table 3.1, it is used by many developer to develop IMS software.

### 3.6 CONCEPTUAL SOFTWARE FLOW CHART

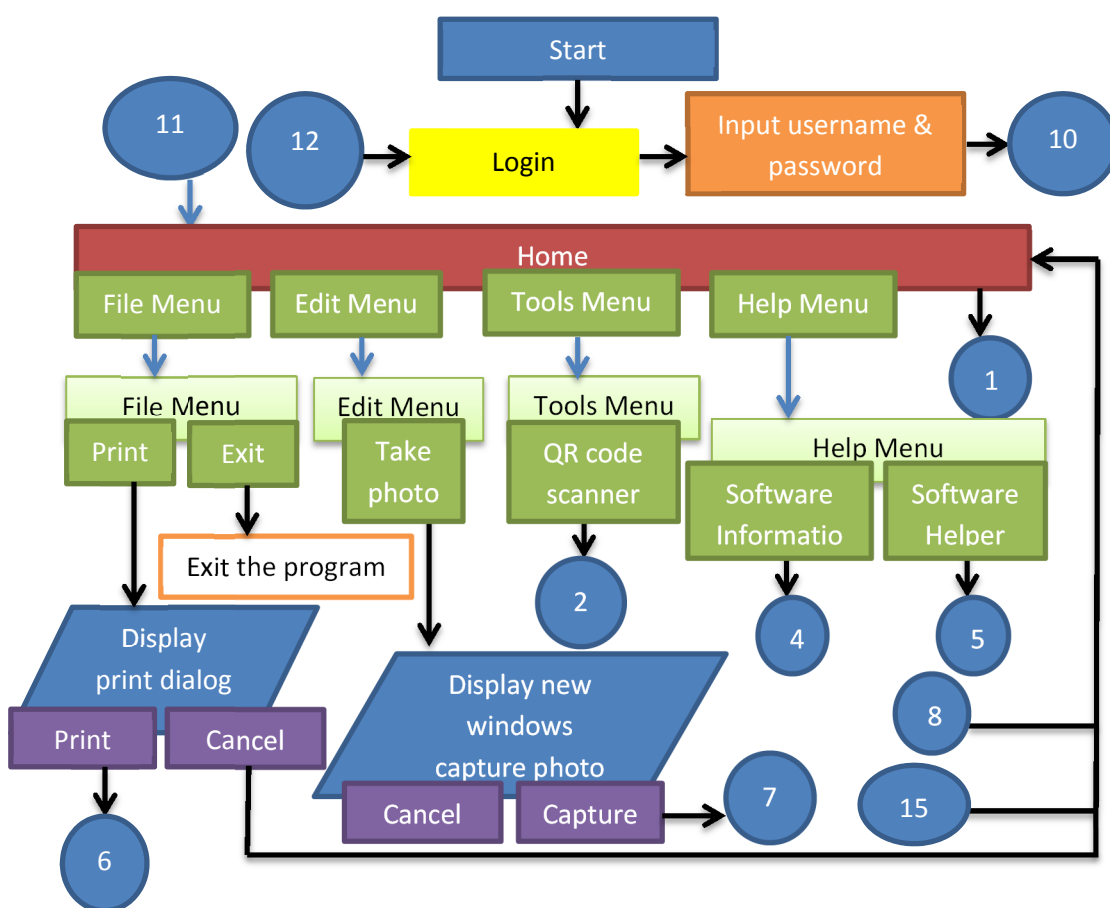


Figure 3.2: Software Flow Chart Part 1

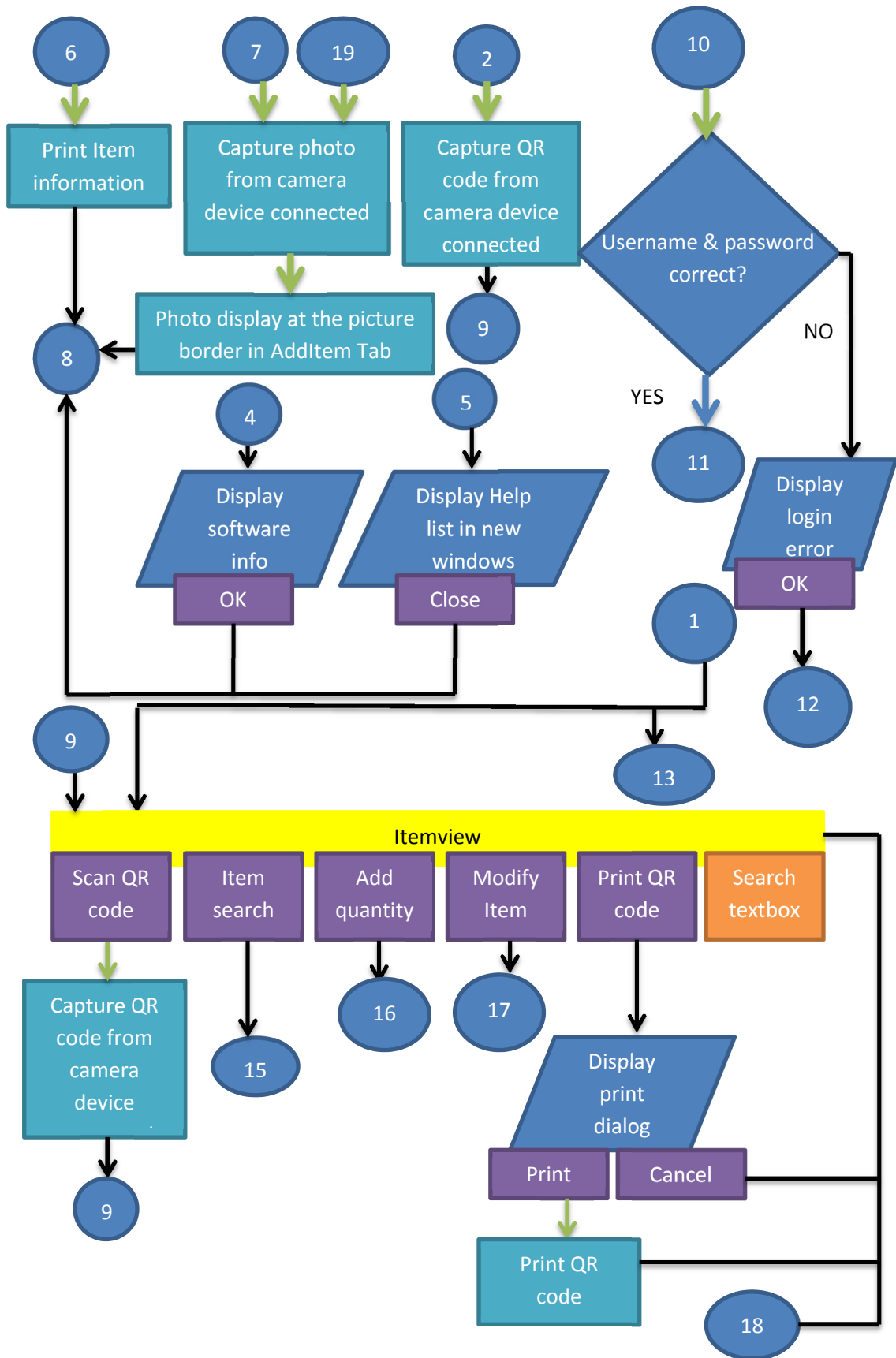
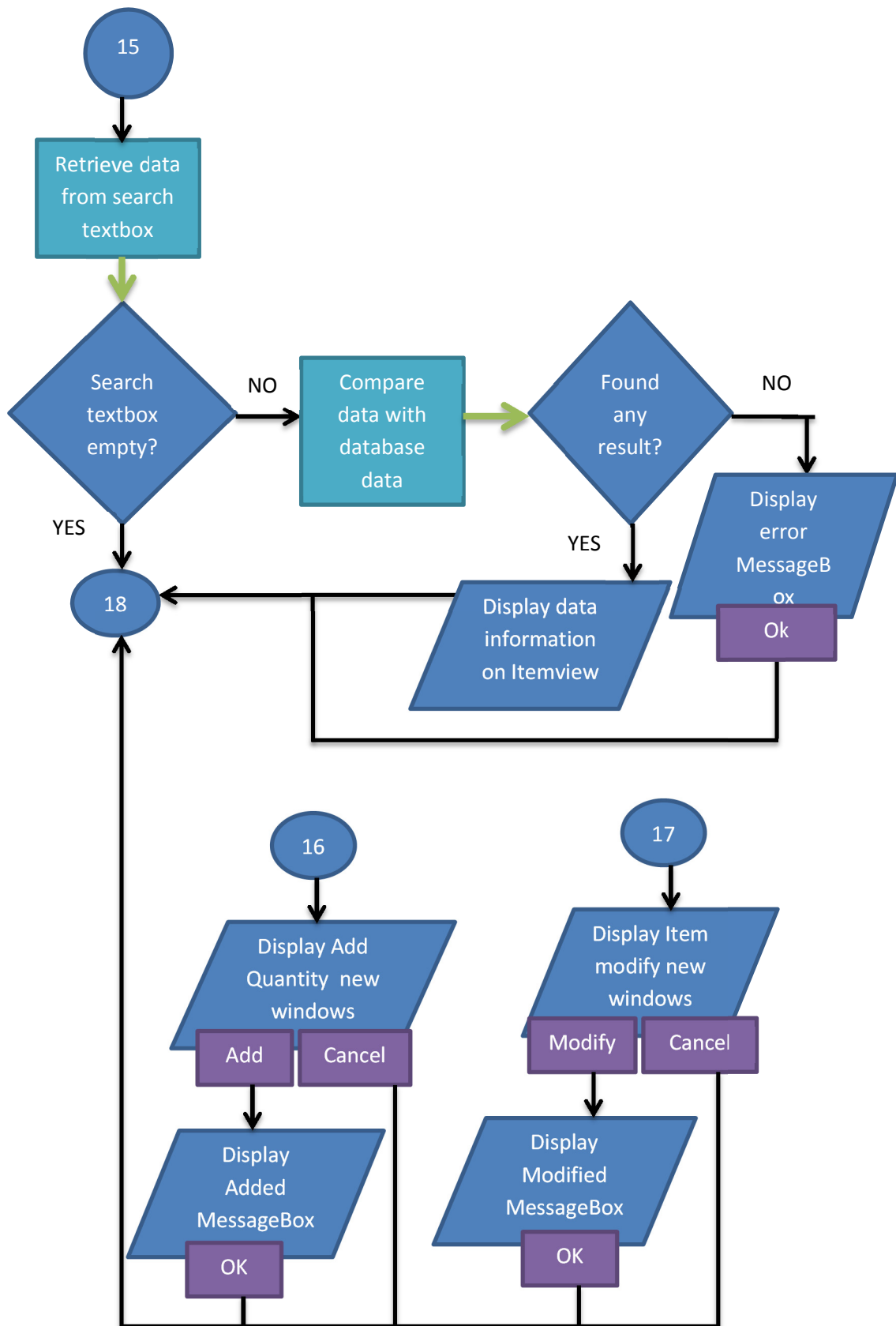


Figure 3.3: Software Flow Chart Part 2





**Figure 3.4:** Software Flow Chart Part 3

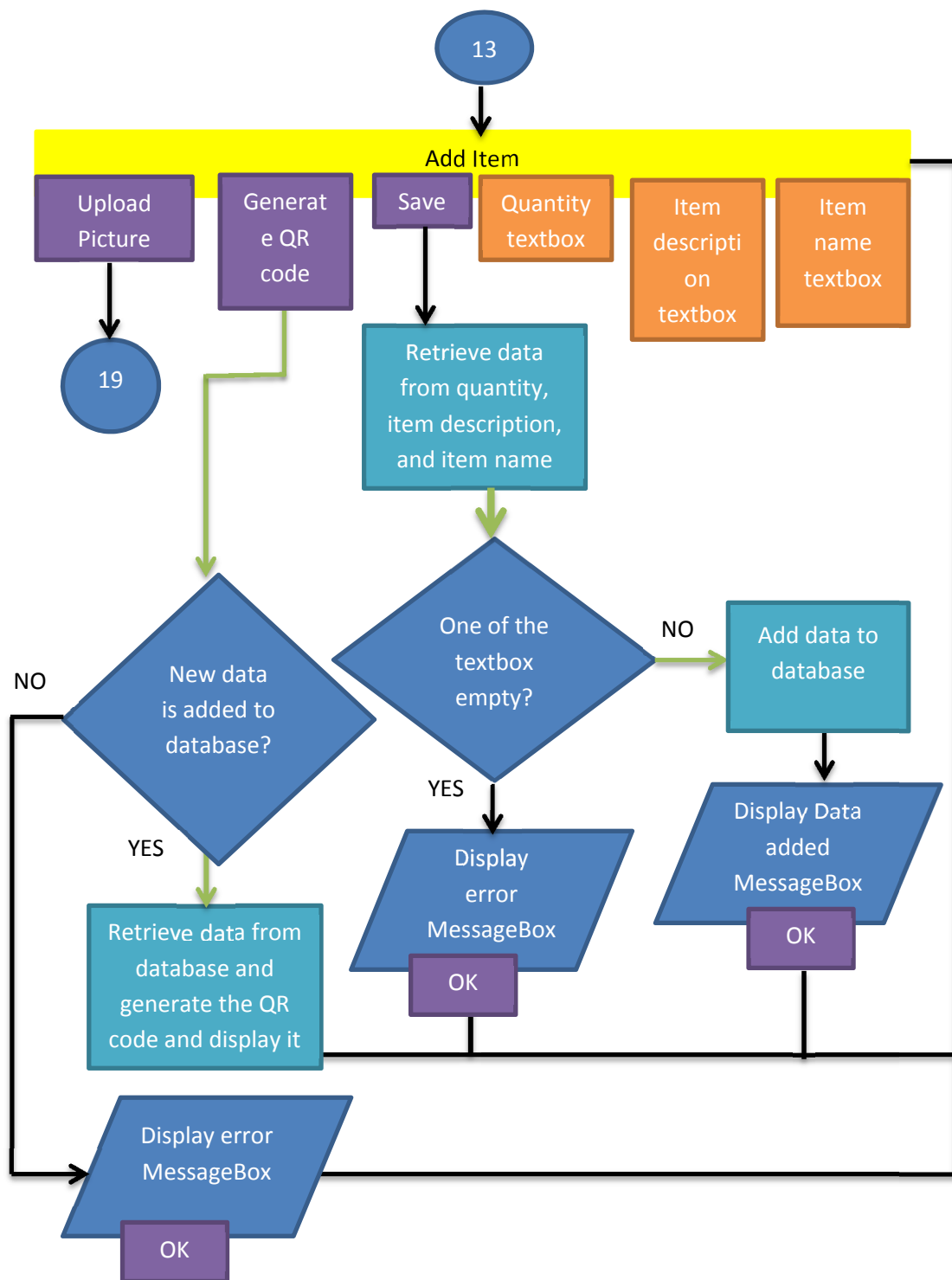
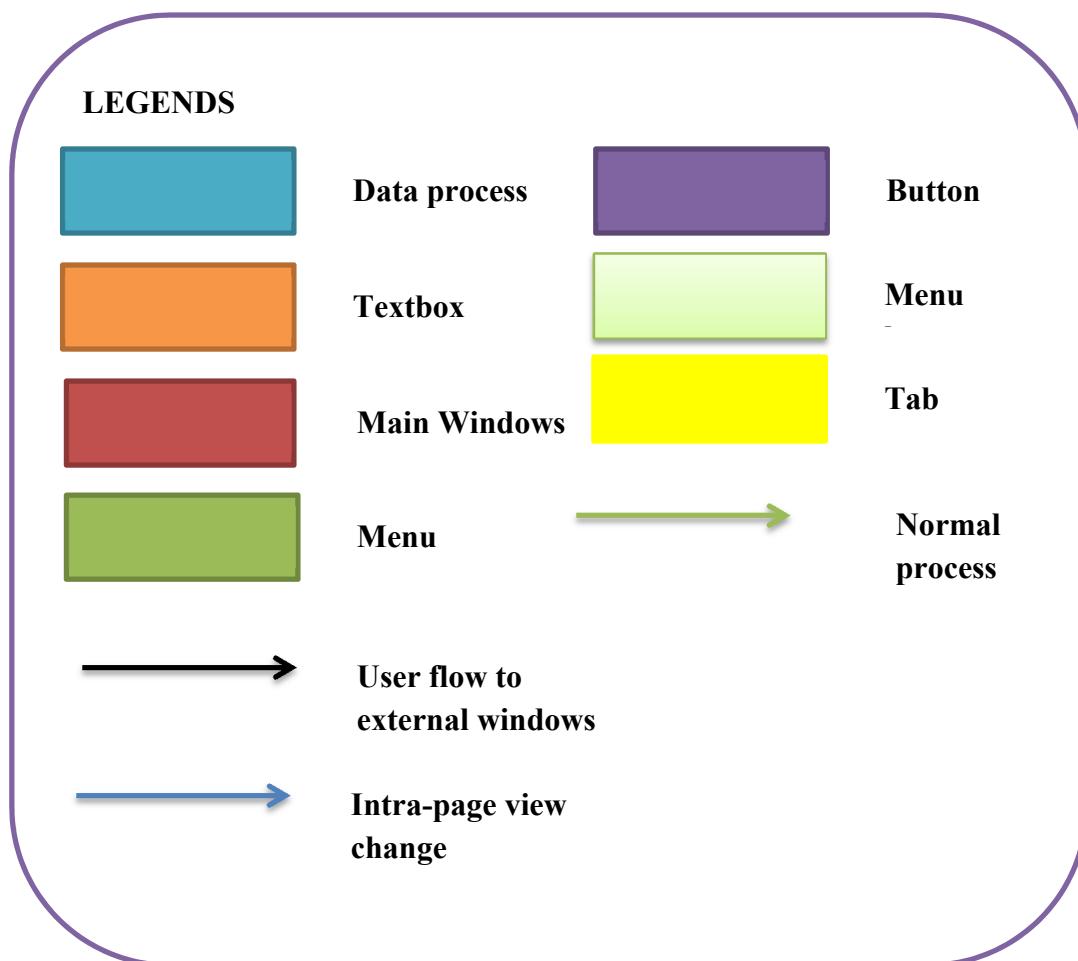


Figure 3.5: Software Flow Chart Part 4

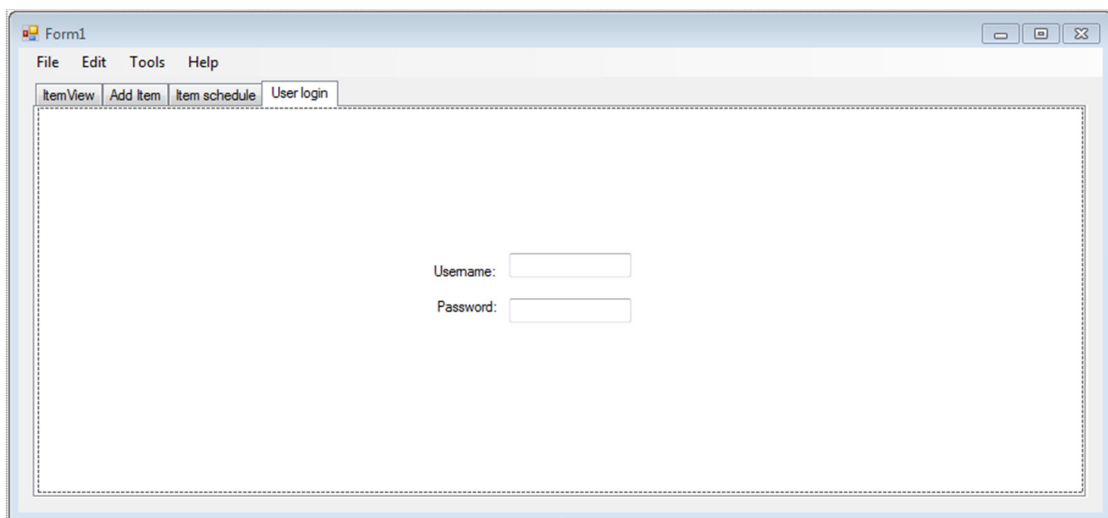


**Figure 3.6:** Software Flow Chart Legends

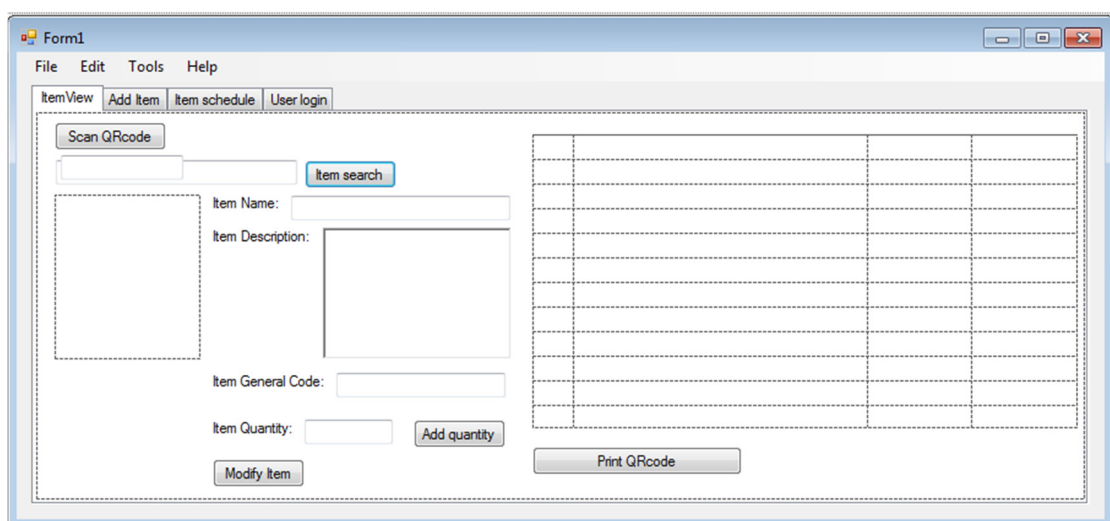
### 3.7 CONCEPTUAL SOFTWARE GUI DESIGN

A conceptual design of the software GUI is done using Visual Studio 2013. It is design using the features and conditions that have set up to visualize the how the end product will looks like. Although the final product GUI maybe different from the conceptual design since function of the GUI have not been added in and debugging have not be done yet. The software conceptual design will have the menu strip where File, Edit, Tools, Help menu is available. The File menu has Print and Exit button. The Edit menu has Take Photo button. Plus, Tools menu has QR code scanner button. Lastly, Help menu has Software Information and Software helper button. First, the Login Tab is where the user key in the username and password to gain access to the other tab. Next, the Itemview Tab enables the user to scan QR code, search item by the name or QR

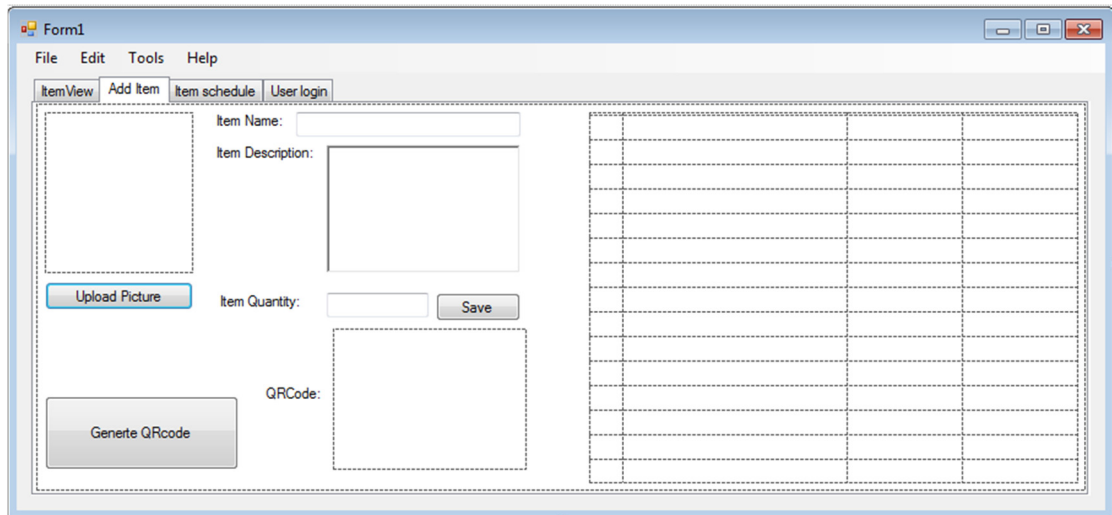
code and view the status and information of the item. In addition, Users are able to modify the item description and print the QR code. Follow up, in Add Item Tab, users are able to add in new Item together with its information and upload the item picture too. Moreover, user can generate QR code by pressing generate QR code button. The QR Code encoder used are from a free source dynamic link library, MessagingToolkit.QRCodec.Codec.dll. The QR Code decoder used zxing to decode and Aforge to communicate with the webcam.



**Figure 3.7:** Login Tab Conceptual Design of GUI

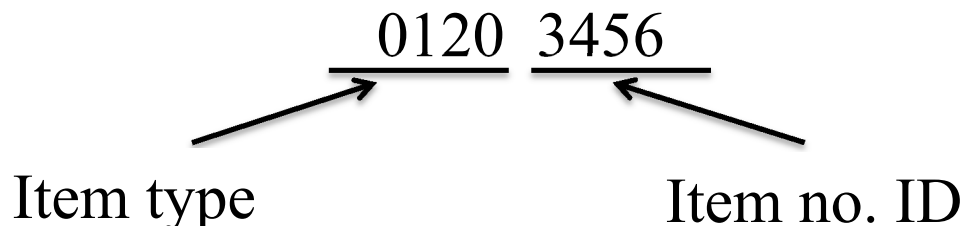


**Figure 3.8:** Itemview Tab Conceptual Design of GUI



**Figure 3.9:** Add item Tab Conceptual Design of GUI

For the QR code, a set of 8 digit number will be used to identify the item unique id. The first 4 number is the identification of the item type and the following 4 number is the identification of the item number. For example, 00104567, 0010 is the number which identify itself as the 10th item register in the database while, 4567 is the identity of the item register under the same item type for the 4567th time.



**Figure 3.10:** QR code identification method

### **3.8 DESIGN THE DATABASE**

The database will be developed using MySQL based xampp. It will be connected to the GUI and hosted as local server using xampp. The database will contain the data from the GUI and the data can be modify through the GUI. Based on the software criteria in section 3.3 and 3.4, the database has to store the item picture, item name, item description, item price, selling price, item check in/out date and time, item code and item quantity.

### **3.9 DEBUG SOFTWARE**

The software will be tested for its functionality and capability and debugging will be carried out by fixing the code of the software until it can function smoothly. Each of the buttons will be tested to ensure it can work properly, and then the QR code scanner function will be tested to ensure it in working state and the reminder function will also be tested.

### **3.10 CONDUCT SURVEY**

The software will be tested by 10 person and a survey form will be created and pass to the 10 person to fill up to test if the user are satisfy with the software. The survey form has 5 question with 5 choices to chosen from which are strongly disagree, disagree, neutral, agree, and strongly agree.

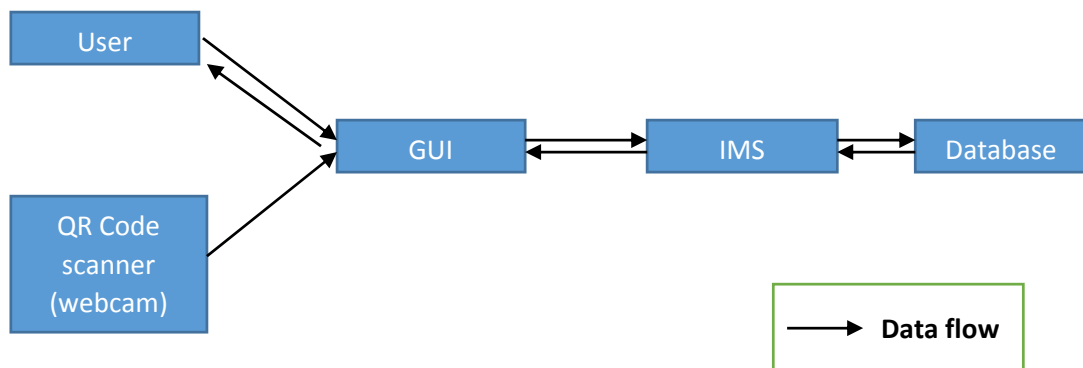
## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 INTRODUCTION

In this section, the project outcome, Inventory Management System (IMS) is discussed and the coding will be explained part by part. Besides that, the problems that are encountered during the process of the software development will be discussed as well.

#### 4.2 OVERVIEW OF IMS SOFTWARE



**Figure 4.1:** Overview of IMS

The IMS can be separated to three main part, Database, IMS and Graphical User Interface (GUI). The database is the place where the data will be stored and used. The GUI is the application to interact with the user, the QR Code and the database.

### 4.3 DEVELOPED DATABASE

The database is built to store all sort of information. It store the user data, general item data and further categorized each item based on the item ID. The database come with username and password to ensure that the data in the database is kept safe from outer interference.

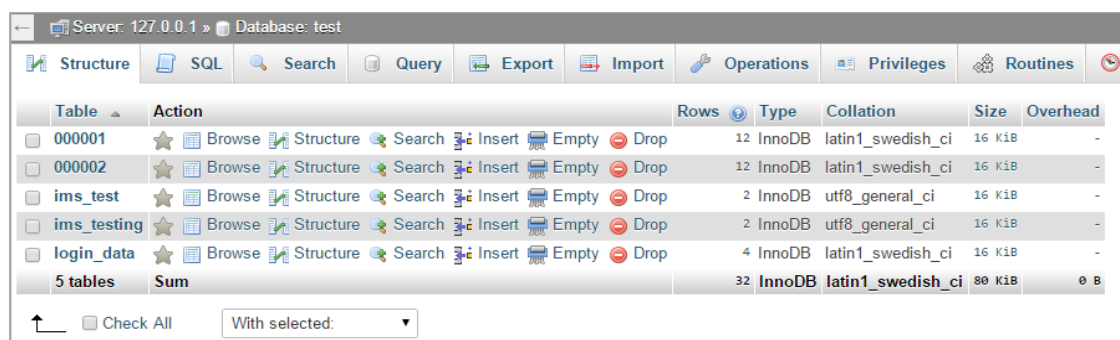
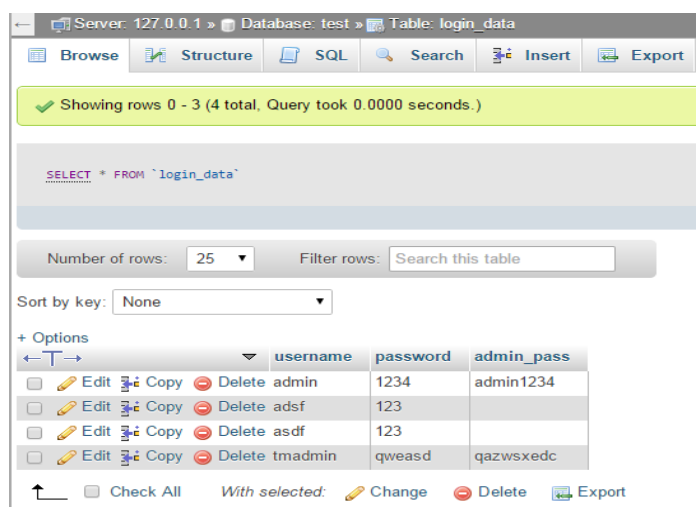


Table	Action	Rows	Type	Collation	Size	Overhead
000001	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
000002	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
ims_test	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16 KiB	-
ims_testing	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16 KiB	-
login_data	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
<b>5 tables</b>	<b>Sum</b>	<b>32</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>88 KiB</b>	<b>0 B</b>

**Figure 4.2:** Overview of the database

The user data is stored in the login\_data, while the general item data such as item name, type, quantity, picture, description, unit price and sold price is store in ims\_test. Then, the item check in and check out detail is store in the item\_id such as 000001 and 000002.



Showing rows 0 - 3 (4 total. Query took 0.0000 seconds.)

```
SELECT * FROM `login_data`
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

Options	username	password	admin_pass
Edit Copy Delete	admin	1234	admin1234
Edit Copy Delete	asdf	123	
Edit Copy Delete	asdf	123	
Edit Copy Delete	tadmin	qweasd	qazwsxedc

**Figure 4.3:** Login\_data data table example



number	id	item_name	item_description	item_quantity	item_qr	item_pic
1	000001	asdf	1212eadds	12		[BLOB - 30.4 KiB]
2	000002	ggger	gerasdadf ajkl dsf adfkl a /V;	12		[BLOB - 30.4 KiB]

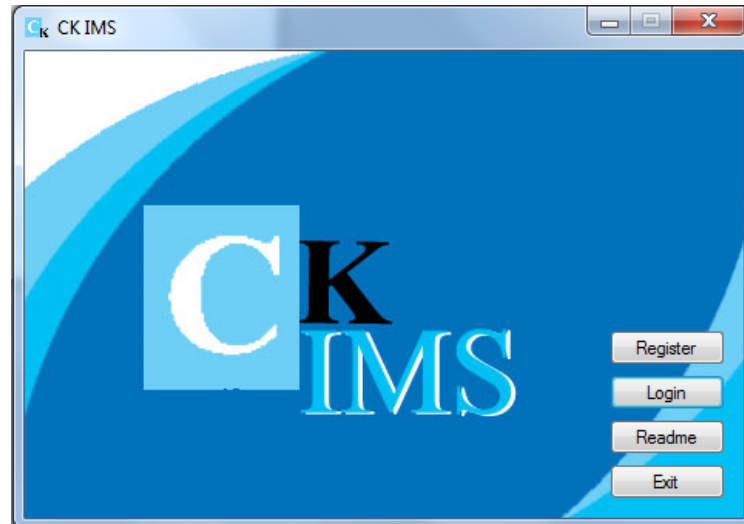
**Figure 4.4:** ims\_test data table example

item_code	check_in	check_out
000002000000	03.04.2015 07:12:50	NULL
000002000001	03.04.2015 07:12:50	NULL
000002000002	03.04.2015 07:12:50	NULL
000002000003	03.04.2015 07:12:50	NULL
000002000004	03.04.2015 07:12:50	NULL
000002000005	03.04.2015 07:12:50	NULL
000002000006	03.04.2015 07:12:50	NULL
000002000007	03.04.2015 07:12:50	NULL
000002000008	03.04.2015 07:12:50	NULL
000002000009	03.04.2015 07:12:50	NULL
000002000010	03.04.2015 07:12:50	NULL
000002000011	03.04.2015 07:12:50	NULL

**Figure 4.5:** 000002 data table example

#### 4.4 DEVELOPED GUI

The GUI can be separated into 6 parts based on their functions which are, user initial selection, data display, search, add item, user register, and check out.



**Figure 4.6:** User selection form

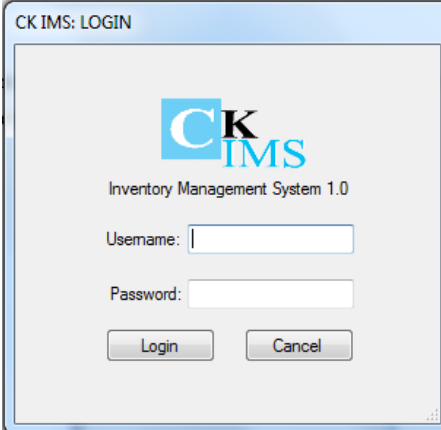
In the figure 4.6, it shows the first form of the application. It allows the user to select register, login, readme and exit. By clicking the register button, a new form named Register will appear and enable the new user to register themselves. For login button, the user will be able to login to the application by filling up their detail in the new form, Login. By clicking readme, a new form, Readme will appear and in it, software origin and the term and condition will be display in a text box.

A screenshot of a Windows application window titled "Register". The window contains the following fields and controls:

- RegisterCode: [text input field]
- New username: [text input field]
- Password: [text input field]
- Confirm Password: [text input field]
- Click to agree to the them and condition
- View term and condition [button]
- Register [button]
- Cancel [button]

**Figure 4.7:** Register form

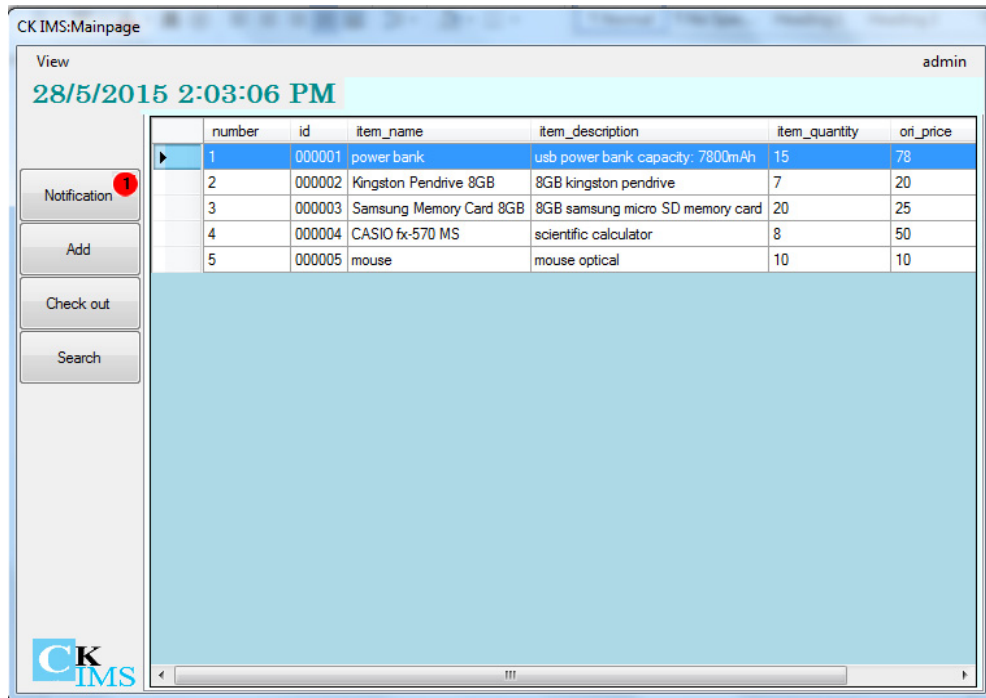
After the detail in register form is put in, the register code is compare with the login data in the database and if it is correct, a new row of data will be added into the login data table in the database. ( Note: the register code is own by the admin, to register a new user, admin has to be notify and help the new user to register a new account for security purpose)



The image shows a screenshot of a web application's login form. The window has a title bar that reads "CK IMS: LOGIN". Inside the window, there is a logo for "CK IMS" where the 'C' and 'K' are in blue and 'IMS' is in black. Below the logo, it says "Inventory Management System 1.0". There are two text input fields: one labeled "Username:" and another labeled "Password:". At the bottom of the form, there are two buttons: "Login" and "Cancel".

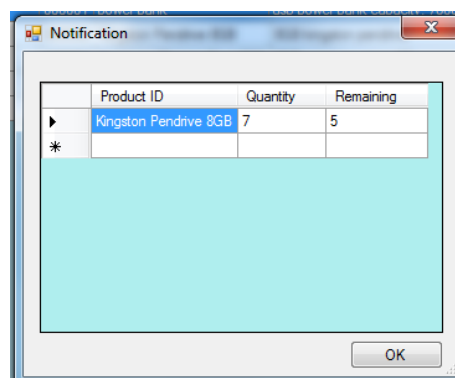
**Figure 4.8:** Login form

After the user has enter the correct username and password, and clicking the login button. The data will be compared with the database login data. If the data enter by user matches with the data in the database. A new form, main page will pop out.



**Figure 4.9:** Mainpage form

The mainpage form display in figure 4.9 is where all the item data is display to the user and the user can use add, search, and check out function as they want to update, view and check the data in the IMS. And the notification button will light up a red colour circle beside it when there are item with less than 5 unit per item. When the notification button is clicked, the form shown in figure 4.10 is displayed. After ok is pressed, the user can select to disable the alarm or leave it the same way.



**Figure 4.10:** Notification form

CK IMS: Item Box

ITEM ID: 000001  
 Item Name: power bank  
 Quantity: 15 /15  
 Description: usb power bank capacity: 7800mAh

Original Price(RM) : 78  
 Selling Price(RM) : 95

Select Printer:   
 print all qrcode

Return

Item List

item_code	check_in	check_out
000001000000	19.05.2015 10:59:37	
000001000001	19.05.2015 10:59:37	
000001000002	19.05.2015 10:59:37	
000001000003	19.05.2015 10:59:37	
000001000004	19.05.2015 10:59:37	

**Figure 4.11:** Item box form

Item box form in figure 4.11 will be displayed when the user had search for the specific item or when the user double click on the item on the item list. It will display the full item code for every item with the check in and check out time of the item. Besides that, user can select the printer and press the print all qrcode button to print all QR Code according to the list in figure 4.11.

CK IMS: Item Addition

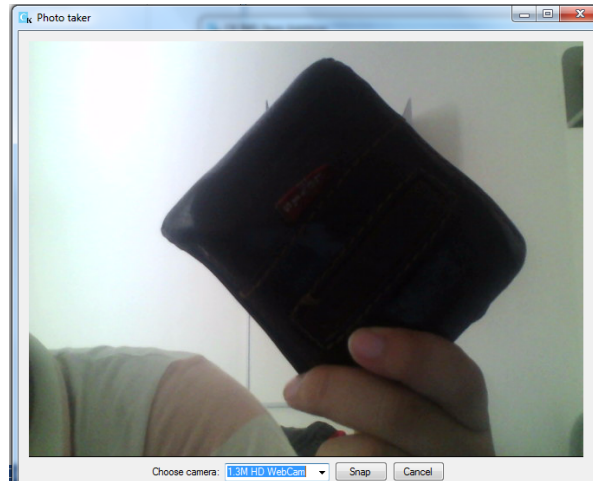
ITEM ID: 000006

Item name:   
 Item description:   
 Original Price(RM) :   
 Item quantity:   
 Selling Price(RM) :   
 Save Cancel

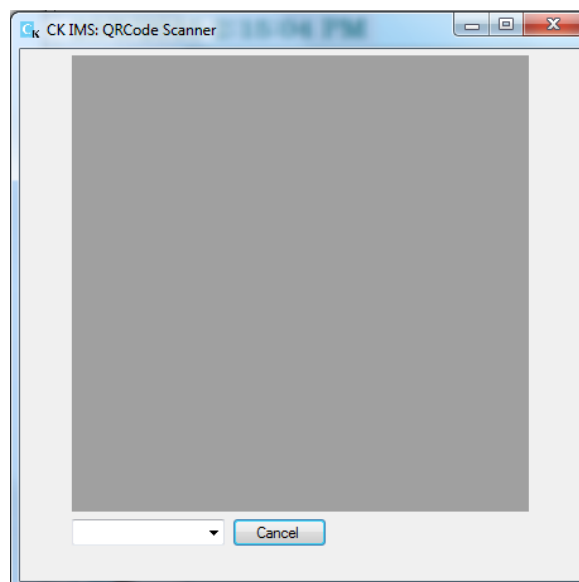
Upload Picture Take photo

**Figure 4.12:** Add item form

In figure 4.12, add item form enable the user to add in new data to the database and auto generate QR Code for each item and display it in Item box form. Moreover, the user can select to choose a picture in the computer or snap a new photo for the item as shown in the figure 4.13.

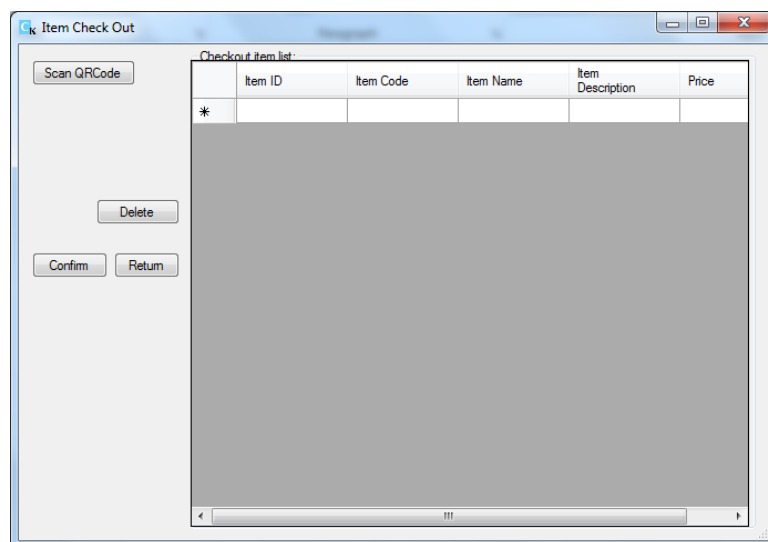


**Figure 4.13:** Photo Taker form



**Figure 4.14:** QR Code scanner form

In figure 4.14, QR Code scanner form will let user to select the image device and capture and decode the QR Code immediately and display the item box form to show the item details. The QR Code Scanner form will appear when the scan QR Code button is clicked.



**Figure 4.15:** Check out form

In figure 4.15, user can press scan QRCode button to scan the QR Code of the item that need to be checked out and the list of the checkout item will be displayed in the list once the QR Code is scanned. If the user want to remove the scanned item from the list, the user can select the item in the list and press Delete button to remove it from the list. After the user have confirmed that the list of item is correct, user can press the Confirm button to confirm the selection and the item will be checked out and the checkout time will be added into the information of that specific item plus the total amount of the item will reduce accordingly.

The developed GUI is different than the conceptual design due to certain coding restrain and for better view of the GUI. First of all, the GUI now is formed with 9 different forms that will be shown to the user based on their selection, this make it more comfortable than having all of it in one form and different tab. Then, the GUI have the initial form to let the user to select between login and register a new account will make it easier to register a new account than have to login in to register a new account. Then

the mainpage form developed is slightly different than the one in conceptual design, it does not have the menu bar of file and instead all the function is list at the left side for the user to pick from. This will let the user find the function that they want without going through every menu bar. Then the selection of the printer and webcam cannot be pre-set by the user due to the limitation of coding for transfer the data from one form to another form. Hence, user has to reselect the webcam and printer once they have changed the form.

#### 4.5 SURVEY CONDUCTED

A survey is carried out to test the user satisfaction on the IMS software developed. 10 subjects is asked to try out the IMS software developed in this project and then fill up the survey form in the appendix. The total score is 194/250 in total and the final result is the subjects are 77.6% satisfy with the software.

people	Question Score				
	Q1	Q2	Q3	Q4	Q5
1 <sup>st</sup>	3	4	4	2	3
2 <sup>nd</sup>	5	5	3	4	4
3 <sup>rd</sup>	4	4	4	3	4
4 <sup>th</sup>	5	5	3	3	5
5 <sup>th</sup>	3	5	4	3	5
6 <sup>th</sup>	3	5	3	2	3
7 <sup>th</sup>	4	4	4	5	5
8 <sup>th</sup>	3	5	3	3	3
9 <sup>th</sup>	3	5	4	4	4
10 <sup>th</sup>	3	5	3	3	4
<b>Total</b>	<b>36</b>	<b>47</b>	<b>35</b>	<b>36</b>	<b>40</b>

**Table 4.1:** Survey result



## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1 INTRODUCTION**

In this section, the conclusion of the project and future recommendation will be discussed. The project outcome has achieved the objective and the scope of the project initially set. The future recommendation will be discussed for further improvement for similar project.

#### **5.2 CONCLUSION**

In, this project, Visual Studio 2013 software development tool that is used to develop the Inventory Management System (IMS) software is the best option out of all the other software development tool as it has a large library and resources and enable coding in many languages. In additional, the Graphical User Interface (GUI) is very user friendly. For the database, the MySQL database is a better choice compare to Microsoft Access and Excel due to it being faster in connection.

The IMS software developed is easy to use and it only need to be setup once for new user. Besides that, it can store a total amount of 999999 items and able to keep the data until it is delete from the database. Moreover, it can print the QR Code for each item and has the low item stock warning feature. The QR Code is scanned using webcam to reduce the cost of the IMS as the barcode scanner price is higher than the webcam however, webcam has its own demerit such as it can't detect QR Code in a low light area.

### **5.3 RECOMMENDATION**

For the future recommendation, this project can be further improved by integrate it with an online server so that the software can be share among other computer and even mobile devices. Besides that, the software can be developed using PHP code or other web script so that it can be make available for anyone in the company that have the register as a user to use the software at any computer in the company. Moreover, additional method of item identification such as RFID or barcode can be added into the system for variation. Besides that, the database can be build using graphical database to store the data more effectively.

## REFERENCE

- Adoga, I., & Valverde, R. (2014). AN RFID Based Supply Chain Inventory Management Solution for the Petroleum Development Industry: A Case Study for Shell Nigeria. *Journal of Theoretical and Applied Information Technology*, 62(1), 199-203.
- Barnhill, B. (2015, May 28). *codeproject*. Retrieved from codeproject, Barcode-Image-Generation-Library: <http://www.codeproject.com/Articles/20823/Barcode-Image-Generation-Library>
- D, L. (2015, May 28). *blogspot*. Retrieved from lekshmiddepu.blogspot: <http://lekshmiddepu.blogspot.com/2012/01/connecting-to-mysql-database-in.html>
- Eclipse. (2015, May 28). *Eclipse*. Retrieved from Eclipse windows Builder: <http://www.eclipse.org/windowbuilder/>
- Gelogo, Y. E., & Kim, H. K. (2014). Development of Mobile Enterprise Inventory Management System Application with CBD.
- Gullaksen, K. (2015, May 28). *github*. Retrieved from github, kenglxn/QRGen: <https://github.com/kenglxn/QRGen>
- Hamid, W., Harun, W., & Loh, C. H. (2008). Development and implementation of radiofrequency identification (RFID) technology for inventory management system: a case study. *Jurnal Mekanikal*, (27), 51-68.
- hellemons, j. (2015, May 28). *Nuget*. Retrieved from Qr Code library 1.3.0: <https://www.nuget.org/packages/MessagingToolkit.QRCode/>
- Herrmann, R. (2015, May 28). *Github*. Retrieved from Github, codebude.QRCoder: <https://github.com/codebude/QRCode>
- Jing, X., & Tang, P. (2013, October). Research and Design of the Intelligent Inventory Management System Based on RFID. In *Computational Intelligence and Design (ISCID), 2013 Sixth International Symposium on* (Vol. 2, pp. 8-11). IEEE.
- Ma, X., & Liu, T. (2011, August). The application of Wi-Fi RTLS in automatic warehouse management system. In *Automation and Logistics (ICAL), 2011 IEEE International Conference on* (pp. 64-69). IEEE.
- Mathaba<sup>1</sup>, S., Dlodlo, N., Williams, Q., & Adigun, M. (2011, April). The use of RFID and Web 2.0 technologies to improve inventory management in South African enterprises. In *Proceedings of the Second International Conference on Information Management and Evaluation* (p. 300). Academic Conferences Limited.
- Microsoft. (2015, May 28). *Microsoft*. Retrieved from Microsoft Library: <https://msdn.microsoft.com/en-us/library/system.windows.forms%28v=vs.110%29.aspx>

- Miyazaki, M. (2015, May 28). *Open Source QR Code Library*. Retrieved from Open Source QR Code Library: <http://qrcode.osdn.jp/index.html.en>
- MySQL. (2015, May 28). *dev.mysql*. Retrieved from mysql connector: <https://dev.mysql.com/downloads/connector/>
- Owen, S. (2015, May 28). *Github*. Retrieved from Github, zxing: <https://github.com/zxing/zxing>
- Rached, E. (2015, May 28). *codeproject*. Retrieved from codeproject, connect-c-to-MySQL: <http://www.codeproject.com/Articles/43438/Connect-C-to-MySQL>
- Su, W., Hu, K., Zhang, L., & Ma, L. (2009, July). A RFID-Based Material Supply Management System in Automatic Vehicle Assembly Streamline. In *Information Technology and Computer Science, 2009. ITCS 2009. International Conference on* (Vol. 2, pp. 259-262). IEEE.
- Tee, K. X., Chew, M. T., & Demidenko, S. (2011, January). An Intelligent Warehouse Stock Management and Tracking System Based on Silicon Identification Technology and 1-Wire Network Communication. In *Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE International Symposium on* (pp. 110-115). IEEE.
- Ting, S. L., & Tsang, A. H. (2012). Design of an RFID-based inventory control and management system: a case study. *The West Indian journal of engineering*, 34(2-Jan), 70-79.
- Vellingiri, S., Ray, A., & Kande, M. (2013, November). Wireless infrastructure for oil and gas inventory management. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE* (pp. 5461-5466). IEEE.
- Wang, W. L., Huang, C. T., Wang, S. J., Chang, C. P., & Liao, H. W. (2013). Application of RFID on equipment parts readiness management system of semiconductor packaging plant. *International Journal of Information Technology and Management*, 12(1), 67-84.
- Wu, Y., Jiang, Z., & Han, N. (2010, January). The design and implementation of the Integrated Supply Chain Management System based on UML and J2EE technology. In *Logistics Systems and Intelligent Management, 2010 International Conference on* (Vol. 3, pp. 1361-1365). IEEE.
- Xia, L. (2011, June). The design and implementation of distributed inventory management system based on the intranet architecture. In *Information and Automation (ICIA), 2011 IEEE International Conference on* (pp. 248-251). IEEE.
- Yeh, K. C., Chen, R. S., & Chen, C. C. (2011). Intelligent service-integrated platform based on the RFID technology and software agent system. *Expert Systems with Applications*, 38(4), 3058-3068.

## APPENDIX A

### IMS SOFTWARE CODING

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IMS_ver1._0
{
    public partial class Userselection : Form
    {
        public Userselection()
        {
            InitializeComponent();
        }

        private void Userselection_Load(object sender, EventArgs e)
        {

        }

        private void Userselection_FormClosed(object sender, FormClosedEventArgs e)
        {
            Application.Exit();
        }

        private void button2_Click(object sender, EventArgs e) //register btn
        {
            Register new_reg = new Register();
            this.Hide();
            new_reg.ShowDialog();
            new_reg = null;
            this.Show();
        }

        private void button1_Click(object sender, EventArgs e) //login btn
        {
            this.Hide();
            Login new_login = new Login();
            new_login.ShowDialog();
        }

        private void button4_Click(object sender, EventArgs e)//exit btn
        {
            Application.Exit();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Readme new_readme = new Readme();
            new_readme.Show();
        }
    }
}
```

```

    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace IMS_ver1._0
{
    public partial class Register : Form
    {
        public Register()
        {
            InitializeComponent();
        }

        private void Register_FormClosed(object sender, FormClosedEventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e) //Cancel button
        {
            this.Close();
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked == true)
            {
                button1.Enabled = true;
            }
            else if (checkBox1.Checked == false)
            {
                button1.Enabled = false;
            }
        }

        private void button1_Click(object sender, EventArgs e) //register button
        {
            bool condition = checkregistercode();
            if (condition ==true && !string.IsNullOrEmpty(textBox2.Text)
&& !string.IsNullOrEmpty(textBox3.Text) && !string.IsNullOrEmpty(textBox4.Text) &&
textBox3.Text.ToUpper().Trim().Equals(textBox4.Text.ToUpper().Trim()))
            {
                MySqlConnection connection = new
MySqlConnection("Server=localhost;Database=test;Uid=root,Password=4foryou");
                connection.Open();
                try
                {
                    MySqlCommand command = connection.CreateCommand();
                    command.CommandText = ("INSERT INTO
login_data(username,password) VALUES(@username,@password)");

```

```

        command.Parameters.AddWithValue("@username", textBox2.Text);
        command.Parameters.AddWithValue("@password", textBox3.Text);
        command.ExecuteNonQuery();
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        if (connection.State == ConnectionState.Open)
        {
            connection.Close();
            MessageBox.Show("registered", "New Register");
            this.Hide();
        }
    }
}
else if
(!textBox3.Text.ToUpper().Trim().Equals(textBox4.Text.ToUpper().Trim()))
{
    MessageBox.Show("Confirm password is incorrect!", "Register Error",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
else if (condition == false)
{
    MessageBox.Show("Register Code is incorrect!", "Register Error",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
else
{
    MessageBox.Show("Please key in new username and
    password!", "Register Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
}

private void Register_Load(object sender, EventArgs e)
{
    textBox1.PasswordChar = '*';
    textBox3.PasswordChar = '*';
    textBox4.PasswordChar = '*';
}

private bool checkregistercode ()
{
    bool condition;
    MySqlConnection connection = new
    MySqlConnection("Server=localhost;Database=test;Uid=root;Password=4foryou");
    connection.Open();
    MySqlCommand command = new MySqlCommand("SELECT * FROM login_data where
    admin_pass ='" + this.textBox1.Text + "'", connection);
    MySqlDataReader data_reader = command.ExecuteReader();

    int count = 0;
    while (data_reader.Read())
    {
        count = count + 1;
    }
    if (count == 1)
    {

```

```

        condition = true;
        return condition;
    }
    else
    {
        condition = false;
        return condition;
    }
}

private void button3_Click(object sender, EventArgs e)
{
    TnC term_cond = new TnC();
    term_cond.ShowDialog();
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Configuration;
using System.IO;

namespace IMS_ver1._0
{
    public partial class Mainpage : Form
    {
        int id;
        int amount;
        int low_alarm;
        string item_id;
        string item_name;
        string item_desc;
        string item_no;
        byte[] item_pic;
        string item_price;
        string item_sell_price;
        int pic_size;
        string user_name;
        string[] itembox_name;
        string[] itembox_quantity;
        string[] itembox_remaining;

        public Mainpage(string username)
        {
            InitializeComponent();
            usernameToolStripMenuItem.Text = username;
            user_name = username;
            timer1.Start();
        }
    }
}

```



```
private void Mainpage_Load(object sender, EventArgs e)
{
    loaddata();
    load_notification();
    if (low_alarm > 0)
    {
        label1.Text = low_alarm.ToString();
        label1.Visible = true;
        pictureBox2.Visible = true;
    }
    else
    {
        label1.Visible = false;
        pictureBox2.Visible = false;
    }
    autocompletetext();
}

private void Tab_DrawItem(object sender, DrawItemEventArgs e)
{
}

private void logoutToolStripMenuItem_Click(object sender, EventArgs e)
{
}

private void printToolStripMenuItem_Click(object sender, EventArgs e)
{
}

private void timer1_Tick(object sender, EventArgs e)
{
    this.datetimetxt.Text = DateTime.Now.ToString();
}

private void Searchbtn_Click(object sender, EventArgs e)
{
    panel2.Visible = true;
    searchtxtbox.Focus();
}

private void Addbtn_Click(object sender, EventArgs e)
{
    this.Hide();
    AddItem adding = new AddItem();
    adding.ShowDialog();
    adding = null;
    this.Show();
    loaddata();
    load_notification();
}

private void searchToolStripMenuItem_Click(object sender, EventArgs e)
```

```

    {
        panel2.Visible = true;
        searchtxtbox.Focus();
    }

    private void button2_Click(object sender, EventArgs e) // scanqr code
    {
        qrscanner qrcode_form = new qrscanner();
        qrcode_form.Show();
        panel2.Visible = false;
    }

    private void Mainpage_FormClosed(object sender, FormClosedEventArgs e)
    {
    }

    private void DataView_CellContentDoubleClick(object sender,
    DataGridViewCellEventArgs e)
    {
    }

    private void DataView_CellDoubleClick(object sender,
    DataGridViewCellEventArgs e)
    {
        id =
        Convert.ToInt32(DataView.Rows[e.RowIndex].Cells["number"].Value.ToString());
        MySqlConnection connection = new
        MySqlConnection("Server=localhost;Database=test;Uid=root,Password=4foryou");
        MySqlCommand command = connection.CreateCommand();
        command.CommandText = "SELECT * FROM ims_test where number = " + id + "";
        DataTable itemtable = new DataTable();
        MySqlDataAdapter display2 = new MySqlDataAdapter(command);
        display2.Fill(itemtable);
        dataGridView1.DataSource = itemtable;
        foreach (DataRow d_row in itemtable.Rows)
        {
            item_id = d_row["id"].ToString();
            item_name = d_row["item_name"].ToString();
            item_desc = d_row["item_description"].ToString();
            item_no = d_row["item_quantity"].ToString();
            item_price = d_row["ori_price"].ToString();
            item_sell_price = d_row["sellin_price"].ToString();
            item_pic = ((byte[])d_row["item_pic"]);
            pic_size = item_pic.GetUpperBound(0);
        }
        Item_box itembox = new Item_box(item_id, item_name, item_desc, item_no,
        item_pic, pic_size, null, item_price, item_sell_price);
        itembox.Show();
    }

    private void loaddata()
    {
        MySqlConnection connection = new
        MySqlConnection("Server=localhost;Database=test;Uid=root,Password=4foryou");
        connection.Open();
        try
        {
            MySqlCommand command = connection.CreateCommand();

```

```

        command.CommandText = "SELECT
number,id,item_name,item_description,item_quantity,ori_price,sellin_price,item_pic,lowstock FROM ims_test";
        MySqlDataAdapter display2 = new MySqlDataAdapter(command);
        DataTable itemtable = new DataTable();
        display2.Fill(itemtable);
        DataView.DataSource = itemtable;
        DataView.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
        DataView.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
        DataView.Columns[7].Visible = false;
        DataView.Columns[8].Visible = false;
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        if (connection.State == ConnectionState.Open)
        {
            connection.Clone();
        }
    }
}

private void load_notification()
{
    int count = DataView.Rows.Count;
    itembox_name = new string[count];
    itembox_quatity = new string[count];
    itembox_remaining = new string[count];
    low_alarm = 0;
    amount = 0;
    string item_code;
    string nameitem;
    int forview = 0;
    bool lowstock = false;
    for (int i = 0; i < DataView.Rows.Count; i++)
    {
        nameitem = DataView.Rows[i].Cells["item_name"].Value.ToString();
        item_code = DataView.Rows[i].Cells["id"].Value.ToString();
        lowstock =
Convert.ToBoolean(DataView.Rows[i].Cells["lowstock"].Value);
        MySqlConnection connection = new
MySqlConnection("Server=localhost;Database=test;Uid=root;Password=4foryou");
        connection.Open();
        try
        {
            MySqlCommand command = connection.CreateCommand();
            command.CommandText = "SELECT item_code, check_in,check_out FROM
`" + item_code + "`";
            MySqlDataAdapter display1 = new MySqlDataAdapter(command);
            DataTable datadisplay = new DataTable();
            display1.Fill(datadisplay);
            dataGridView2.DataSource = datadisplay;
            dataGridView2.AutoSizeRowsMode =
DataGridViewAutoSizeRowsMode.AllCells;

```

```

        dataGridView2.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        if (connection.State == ConnectionState.Open)
        {
            connection.Clone();
        }
    }
    for (int j = 0; j < (dataGridView2.Rows.Count); j++)
    {
        string ans =
dataGridView2.Rows[j].Cells["check_out"].Value.ToString();
        if (String.IsNullOrEmpty(ans))
        {
            amount = amount + 1;
        }
        else
        {
        }
    }
    if (amount < 6 && lowstock == false)
    {
        low_alarm = low_alarm + 1;
        itembox_name[forview] = nameitem;
        itembox_quatity[forview] = dataGridView2.Rows.Count.ToString();
        itembox_remaining[forview] = amount.ToString();
        forview = forview + 1;
    }
    amount = 0;
}

if (low_alarm > 0)
{
    label1.Text = low_alarm.ToString();
    label1.Visible = true;
    pictureBox2.Visible = true;
}
else
{
    label1.Visible = false;
    pictureBox2.Visible = false;
}

}

private void resultbtn_Click(object sender, EventArgs e)
{
    int count = 0;
    int number = 0;
    string searchdata = searchtxtbox.Text;
    for (int i = 0; i < DataView.Rows.Count; i++)
    {

```

```

        if
(searchdata.Equals(DataView.Rows[i].Cells["item_name"].Value.ToString()))
        {
            count++;
            number = i;
        }
        else
        {

        }
    }
    if (count > 0)
    {
        MessageBox.Show("Found matching item", "Search Result",
MessageBoxButtons.OK);
        DataView.CurrentCell = DataView.Rows[number].Cells[0];
    }
    else
    {
        MessageBox.Show("The item is not found in the database, please
search again", "Search Error!!!", MessageBoxButtons.OK);
    }
    resetsearch();
}

private void resetsearch()
{
    searchtxtbox.Text = null;
    panel2.Hide();
}

private void autocompletetext()
{
    var source = new AutoCompleteStringCollection();
    MySqlConnection connection = new
MySqlConnection("Server=localhost;Database=test;Uid=root,Password=4foryou");
    connection.Open();
    try
    {
        MySqlCommand command = connection.CreateCommand();
        command.CommandText = "SELECT item_name FROM ims_test";
        MySqlDataAdapter display3 = new MySqlDataAdapter(command);
        DataSet itemname = new DataSet();
        display3.Fill(itemname);
        display3.Dispose();
        connection.Close();
        foreach (DataRow row in itemname.Tables[0].Rows)
        {
            source.Add(row[0].ToString());
        }
        searchtxtbox.AutoCompleteCustomSource = source;
        searchtxtbox.AutoCompleteSource = AutoCompleteSource.CustomSource;
    }
    catch (Exception)
    {
        throw;
    }
    finally
    {
        if (connection.State == ConnectionState.Open)

```

```

        {
            connection.Clone();
        }
    }

private void searchcncl_Click(object sender, EventArgs e)
{
    resetsearch();
}

private void logoutToolStripMenuItem1_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Are you sure you want to
logout?", "Logout Warning", MessageBoxButtons.YesNo);
    if (result == DialogResult.Yes)
    {
        this.Close();
        Login new_login = new Login();
        new_login.Show();
    }
    else { }
}

private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Are you sure you want to exit?",
"Application Exit Warning", MessageBoxButtons.YesNo);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
    else { }
}

private void Checkoutbtn_Click(object sender, EventArgs e)
{
    this.Hide();
    Checkout_list checkout_page = new Checkout_list();
    checkout_page.ShowDialog();
    checkout_page = null;
    this.Show();
    loaddata();
    load_notification();
}

private void Notifybtn_Click(object sender, EventArgs e)
{
    int number = DataView.Rows.Count;
    Notification new_notification = new Notification(itembox_name,
itembox_quatity, itembox_remaining, number);
    new_notification.ShowDialog();
    new_notification = null;
    DialogResult result = MessageBox.Show("Do you want to disable the
alarm?", "Disable Alarm", MessageBoxButtons.YesNo);
    if (result == DialogResult.Yes)
    {
        for (int i = 0; i < number; i++)
        {
            if (string.IsNullOrEmpty(itembox_name[i]))

```



```

{
    public partial class Item_box : Form
    {
        string item_id;
        string full_item_id;
        string item_na;
        string item_d;
        string item_num;
        string price;
        string sold_price;
        byte[] item_p;
        int picsize;
        string code;
        int amount;
        int itemcount = 0;

        public Item_box(string id, string item_name, string item_desc, string
item_no,byte[] item_pic, int pic_size,string full_id,string ori_price,string
sell_price)
        {
            InitializeComponent();
            item_id = id;
            full_item_id = full_id;
            item_na = item_name;
            item_d = item_desc;
            item_num = item_no;
            item_p = item_pic;
            picsize = pic_size;
            price = ori_price;
            sold_price = sell_price;
            LoadData();
            fixquantitytxt.Text = "/" + item_num;
            load_selection();
            foreach (string printer in
System.Drawing.Printing.PrinterSettings.InstalledPrinters)
            {
                comboBox1.Items.Add(printer);
            }
        }

        private void Item_box_Load(object sender, EventArgs e)
        {
            textBox3.Text = item_id;
            textBox1.Text = item_na;
            textBox2.Text = Convert.ToString(amount);
            richTextBox1.Text = item_d;
            textBox4.Text = price;
            textBox5.Text = sold_price;
            MemoryStream stream = new MemoryStream(item_p,0,picsize);
            pictureBox1.Image = Image.FromStream(stream);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void generateQRcode (string LETTER)

```



```

    {
        QRCodeEncoder encoder = new QRCodeEncoder();
        encoder.QRCodeEncodeMode = QRCodeEncoder.ENCODE_MODE.BYTE; //set the
mode of encoding;
        encoder.QRCodeErrorCorrect = QRCodeEncoder.ERROR_CORRECTION.L; // set
correction level low for more space
        encoder.QRCodeVersion = 7; //set version
        encoder.QRCodeScale = 4; //set size
        Bitmap qrcode = encoder.Encode(LETTER);
        pictureBox2.Image = qrcode as Image;
    }

    private void LoadData()
    {
        MySqlConnection connection = new
MySqlConnection("Server=localhost;Database=test;Uid=root;Password=4foryou");
        connection.Open();
        try
        {
            MySqlCommand command = connection.CreateCommand();
            command.CommandText = "SELECT item_code, check_in,check_out FROM `"+
+ item_id +"`";
            MySqlDataAdapter display1 = new MySqlDataAdapter(command);
            DataTable datadisplay = new DataTable();
            display1.Fill(datadisplay);
            dataGridView1.DataSource = datadisplay;
            dataGridView1.AutoSizeRowsMode =
DataGridViewAutoSizeRowsMode.AllCells;
            dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
        }
        catch(Exception)
        {
            throw;
        }
        finally
        {
            if (connection.State == ConnectionState.Open)
            {
                connection.Clone();
            }
        }
        for (int i = 0; i < (dataGridView1.Rows.Count); i++ )
        {
            string ans =
dataGridView1.Rows[i].Cells["check_out"].Value.ToString();
            if(String.IsNullOrEmpty(ans))
            {
                amount = amount + 1;
            }
            else
            {
            }
        }
    }
}

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)

```

```

        {
            code =
dataGridView1.Rows[e.RowIndex].Cells["item_code"].Value.ToString();
            generateQRcode(code);
            label5.Visible = true;
        }

        private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            code =
dataGridView1.Rows[e.RowIndex].Cells["item_code"].Value.ToString();
            generateQRcode(code);
            label5.Visible = true;
        }

        private void load_selection ()
        {
            if (string.IsNullOrEmpty(full_item_id))
            {

            }
            else
            {
                int count =0;
                int number = 0;
                dataGridView1.ClearSelection();
                for (int i =0; i<(dataGridView1.Rows.Count); i++)
                {
                    if
(full_item_id.Equals(dataGridView1.Rows[i].Cells["item_code"].Value.ToString()))
                    {
                        number = i;
                        count++;
                    }
                    else
                    {

                    }
                }
                if (count>0)
                {
                    dataGridView1.CurrentCell = dataGridView1.Rows[number].Cells[0];
                }
                else
                {
                    MessageBox.Show("error", "error", MessageBoxButtons.OK);
                }
            }
        }

        private void printallbtn_Click(object sender, EventArgs e)
        {
            if (comboBox1.Text != null)
            {
                itemcount = 0;
                PrintDocument pd = new PrintDocument();
                pd.PrintPage += Printpage;
                pd.PrinterSettings.PrinterName = comboBox1.Text;
                pd.Print();
            }
        }

```

```

    }
    else
    {
        MessageBox.Show("Please select printer", "Printing Error",
        MessageBoxButtons.OK);
    }
}

private void Printpage (object o, PrintPageEventArgs e)
{
    int totalpagecount = 0;
    int curr_pagecount = 1;
    int heightcount = 0;
    int width = 50;
    int height = 50;
    totalpagecount = dataGridView1.Rows.Count / 12;
    if ((dataGridView1.Rows.Count % 12) > 0)
    {
        totalpagecount = totalpagecount + 1;
    }
    int count = 0;
    while( count < 12 && curr_pagecount <= totalpagecount && itemcount <
    dataGridView1.Rows.Count)
    {
        code =
        dataGridView1.Rows[itemcount].Cells["item_code"].Value.ToString();
        generateprintQRcode(code);
        e.Graphics.DrawImage(pictureBox3.Image, width, height);
        width = width + pictureBox3.Width + 100;
        heightcount++;
        if (heightcount == 3)
        {
            height = height + pictureBox3.Height + 100;
            heightcount = 0;
            width = 50;
        }
        count++;
        itemcount++;
    }
    if (count == 12 && itemcount < dataGridView1.Rows.Count)
    {
        count = 0;
        curr_pagecount++;
        e.HasMorePages = true;
        width = 50;
        height = 50;
    }
    else
    {
        e.HasMorePages = false;
    }
}

private void generateprintQRcode(string LETTER)
{
    QRCodeEncoder encoder = new QRCodeEncoder();
    encoder.QRCodeEncodeMode = QRCodeEncoder.ENCODE_MODE.BYTE; //set the
mode of encoding;
}

```

```
        encoder.QRCodeErrorCorrect = QRCodeEncoder.ERROR_CORRECTION.L; // set
correction level low for more space
        encoder.QRCodeVersion = 7; //set version
        encoder.QRCodeScale = 4; //set size
        Bitmap qrcode = encoder.Encode(LETTER);
        pictureBox3.Image = qrcode as Image;
    }
}
}
```

## APPENDIX B

### EXAMPLE OF SURVEY FORM



Faculty of Manufacturing Engineering,  
University Malaysia Pahang,  
26600 Pekan,  
Pahang Darul Makmur,  
MALAYSIA  
Phone: 609-4245800 Fax: 609-4245800

#### Development of a Computerized Inventory Management System (IMS) for Industry Application

Principle Investigator: Dr Nizam Uddin Ahamed

Co-Investigator: Ooi Choon Kheng

##### Customer Satisfaction Form on IMS Software

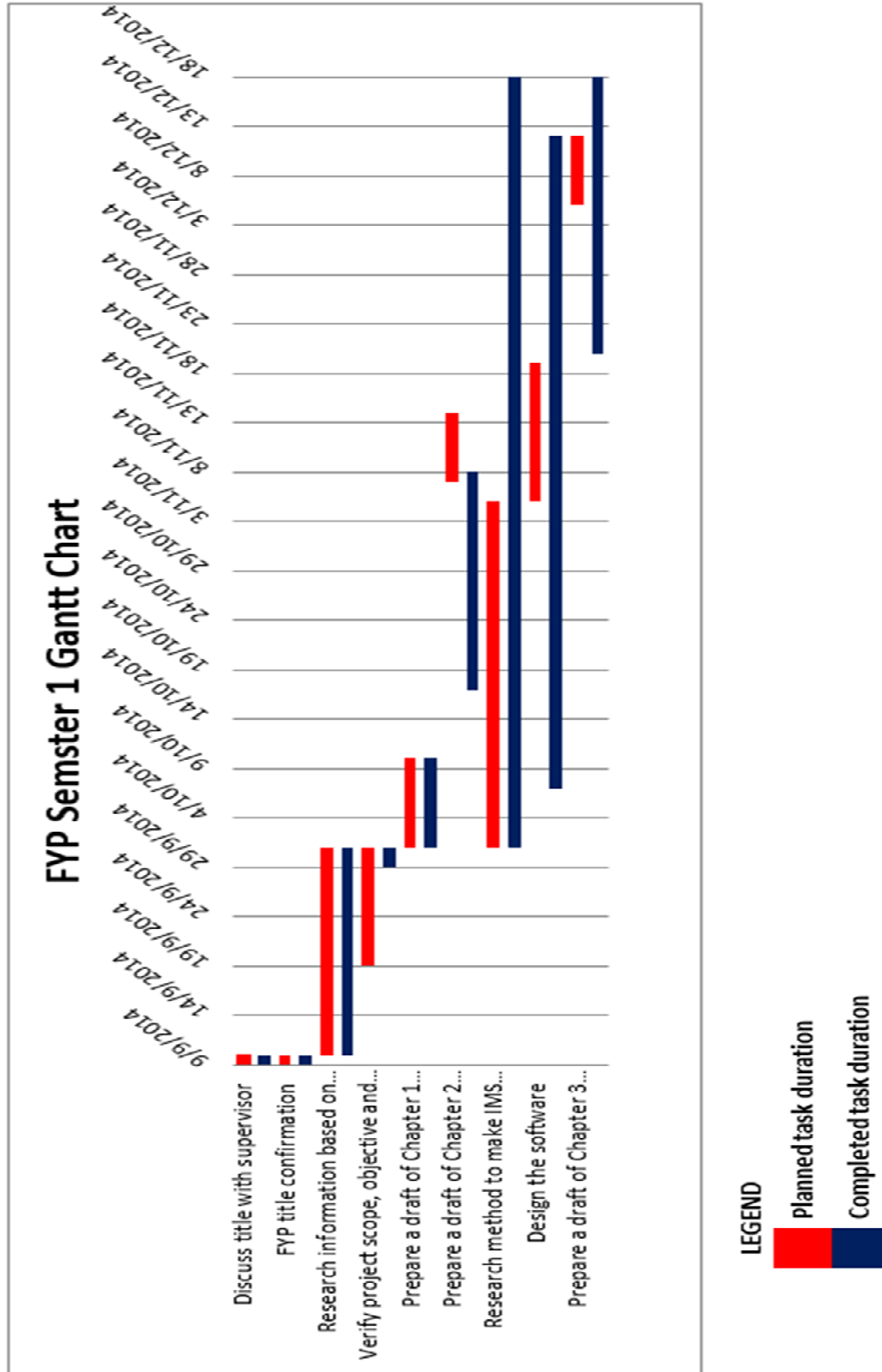
Name: \_\_\_\_\_

Age: \_\_\_\_\_

1. Do you like the looks of the Graphical User Interface (GUI)?  
 Strongly Disagree    Disagree    Neutral    Agree    Strongly Agree
2. Is the GUI convenient for usage?  
 Strongly Disagree    Disagree    Neutral    Agree    Strongly Agree
3. Is the low stock warning useful?  
 Strongly Disagree    Disagree    Neutral    Agree    Strongly Agree
4. Would you use this software to manage your inventory?  
 Strongly Disagree    Disagree    Neutral    Agree    Strongly Agree
5. Is the checkout system convenient?  
 Strongly Disagree    Disagree    Neutral    Agree    Strongly Agree

APPENDIX C1

FYP SEMSTER 1 GANTT CHART



## APPENDIX C2

### FYP SEMSTER 2 GANTT CHART

