



UNIVERSITI
TEKNOLOGI
PETRONAS

Dissertation

Droid Queue Management System: Counter Service
For Queue Management System

By

Nasiratul Safiah binti Mohammad Rushdi

September 2013

TABLE OF CONTENT

LIST OF FIGURE	v
LIST OF TABLE	vi
CHAPTER 1: INTRODUCTION	1
1.1 Background of Study	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope of Study	3
1.5 Relevancy of Project	4
1.6 Summary	4
CHAPTER 2: LITERATURE REVIEW	5
2.1 Queue Definition	5
2.2 Queue (Abstract Data Type)	5
2.3 Queue Area	6
2.4 QLess Mobile Wait Management	7
2.5 Queue Algorithm	8
CHAPTER 3: METHODOLOGY	11
3.1 System Development Methodology	11
3.2 Planning	12
3.3 Analysis	12
3.4 Design	13
3.5 Development	13
3.6 System Architecture	15
3.7 Integration and Test	16
3.8 Implementation	16
3.9 Project Activities	17
3.10 Key Milestone	19
3.10.1 Final Year Project I	19
3.10.2 Final Year Project II	20
3.10.3 Gantt Chart	21
3.11 Tools	22
3.11.1 Software	22
3.11.2 Hardware	22
CHAPTER 4: RESULT & DISCUSSION	23
4.1 Introduction	23

4.2	Data Gathering	23
	4.2.1 Survey Analysis	23
4.3	Prototyping	29
4.4	Pilot Test Result	30
	4.4.1 Counter Service Functionalities	35
CHAPTER 5: CONCLUSION & FUTURE WORK		38
5.1	Conclusion	38
5.2	Future Work	39
REFERENCES		40
APPENDICES		42

LIST OF FIGURES

Figure 1: Represent of a Queue with FIFO (First-In-First-Out) processing	6
Figure 2: Average transaction length	8
Figure 3: Array-based queue	9
Figure 4: Circular array	9
Figure 5: Linked queue	10
Figure 6: Rapid Application Development (RAD) Model	11
Figure 7: Flow of the DQMS	13
Figure 8: Droid Queue Management System Architecture	15
Figure 9: Number of clients per day	24
Figure 10: Smartphone ownership	25
Figure 11: Current Queue Management Feedback	26
Figure 12: Current Queue Management Issues	26
Figure 13: Improvement on Queue Management System	27
Figure 14: Familiarity of mobile apps in QMS	27
Figure 15: Familiarity of green IT usage	28
Figure 16: The importance of green IT products	28
Figure 17: Number requested by the user	31
Figure 18: Display board shows the pending number	32
Figure 19: Counter prompted the user to process the number	32
Figure 20: Counter shown that the number is being processed	33
Figure 21: Display board shown the number 65 is currently being served	33
Figure 22: Counter prompted the next number to be processed	34
Figure 23: Counter retrieves queue number	35
Figure 24: The user accepts the number	35
Figure 25: The service is done	36
Figure 26: Displays the number that hasn't received the service	36
Figure 27: Displays the number that being served	37

LIST OF TABLES

Table 1: Final Year Project I milestone	19
Table 2: Final Year Project II milestone	20

CHAPTER 1

INTRODUCTION

1.1 Background Study

Queue Management has been an issue for a long time in many organizations especially in banks and government departments. Due to the growing technology nowadays, it is not only important to provide convenience to the public consumers, but to help the person behind the counters to manage the existing queue so that the service will be delivered efficiently and effectively. Droid Queue Management System (DQMS) is invented to address the problem in managing queue in the organizations thus will improve the satisfaction of the public consumers as well as the organizations. DQMS will provide the consumers with a digital number, instead of paper ticket number, which is synchronized with the counter system. This study will be focusing on the organizations side on managing the queue from behind the counters.

DQMS is an android application to organize the waiting list exists in the organization. The system will capture the number that is generated by the number station and will then store the list number which to be distributed to the counters and users' smart phones. In addition, the system will synchronize the number updates with the counters and users' smart phones, so that they will keep updated with the current queue list, thus smoothen the process of service.

This application will be developed by using android development tool which is android SDK and eclipse. Instead of using desktop, the counters will be using tablets which can connect with the system in Android operating system.

1.2 Problem Statement

During the peak hours in usual business days, a long waiting line can be so much troublesome to the consumers. Waiting is not only irritating, but can be very tiring to both public consumers and staffs in charged behind the counters. Lately, the problem with waiting lines took center stage when the consumers complained about the service delivered by the staffs were unpleasant and annoying, especially at the last moment of business hour. This problem might happen because of the staffs were too tired managing a big number of crowds every day.

In the following study, the queue at a local bank outlet is traditionally managed by the organization using paper ticket. Customers sit in the waiting area and are entertained by newspapers, magazines, television or just good conversation. However, at high populated area, the crowd is merely hard to manage and the customers packed up in the waiting area. Some customers would leave the outlets momentarily and another customer entered the outlet while they were gone. The later customer would assume that the earlier customer is after them in the waiting list. This would cause an irritable confrontation depending on the individuals involved thus tarnishing the image of the bank.

Sometimes, due to the long queue, there will be a modifiable sign at the start of the queue or halfway point in the queue informing people approximately how long their wait will be. This shows that long queues are hard to manage especially if the organization is having lack of man power. Long queue means a lot of people waiting and it means extra work to the staffs in charged.

1.3 Objectives

Droid Queue Management System (DQMS) is an alternative and a potential revolutionary system to address the problem of the current manual queue ticketing system used in most organizations. The objectives of this project are:

- To design and develop a comprehensive Android based queue management application.
- To evaluate user's perception on the developed application.

1.4 Scope of Study

In order to achieve the objectives through the development of an eco-friendly android application, Droid Queue Management System, a detailed research regarding to the different scopes of study need to be conducted. This system aims to be applied in any organizations which require queuing for the most business activities. The scopes of study throughout the development of the system are as follows:

- Knowledge acquiring and comprehension on the development of mobile native application particularly Android.
- A comprehension on the conventional queue management system in the targeted organizations.
- Integration of different Android devices to simulate the Droid Queue Management System.

1.5 Relevancy of Project

In reference to section 1.2, Droid Queue Management System (DQMS) serves as a solution to the previously discussed problem – current queue management systems are not comprehensive enough. Hence, by researching this domain, queue management system can potentially be improved with reduced time and cost.

1.6 Summary

In this project, an eco-friendly application for Android users to both customers and organizations, called Droid Queue Management System is proposed. This application is an alternative to the current queue management system which provides convenience and improves efficiency in the system. It definitely helps the organizations to increase the customer satisfaction thus enhance the image of the organizations.

The project will be developed using Android mobile platform. There is over millions of smartphones and devices throughout the world which using Android operating systems to run their devices. It is the largest mobile platform and the number is still growing every day. Compare to the other operating system like IOS and Windows phone, Android has the most apps, games and other digital content (Google Inc, 2013).

CHAPTER 2

LITERATURE REVIEW

2.1 Queue Definition

A queue can be defined in different ways. One definition of a queue is, “A particular data type or collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position, known as enqueue, and removal of entities from the front terminal position, known as dequeue. This makes the queue a First-IN-First-Out (FIFO) data structure.” This type of queue is an example of a linear data structure, or more abstractly a sequential collection. Another definition of queue is, “A line of people or vehicles waiting for something.” Both definitions are relevant to this project and will be elaborated in this review.

According to FIFO data structure, the first element added to the queue will be the first one to be withdrawn. This is parallel to the requirement that whenever an element is added, all elements that were added before have to be withdrawn before the new element can be withdrawn.

2.2 Queue (Abstract Data Type)

Queues are implemented as data structures paired with access routines, as an abstract data structure or in object-oriented language as classes. Queues enable various entities such as data, objects, persons, or events are stored and held to be processed later. The queue performs functions of a buffer. Common applications are circular buffers and linked lists.

In another terms, queue is a First-I-First-Out (FIFO) data structure. FIFO is related to ways of handling and manipulation of data relative to time and prioritization. Wikipedia stated that, “... the principle of a queue processing technique or servicing conflicting demands by ordering process by first-come, first-served behavior ...” Some of the real-world examples which use queues are escalators, coin dispensers and vending machines. In each cases, the person or object at the front of the line was the first to enter, while at the end of the line is the last to have entered. This reflects the enqueue and dequeue function, every time another person or object enters and leaves the queue line.

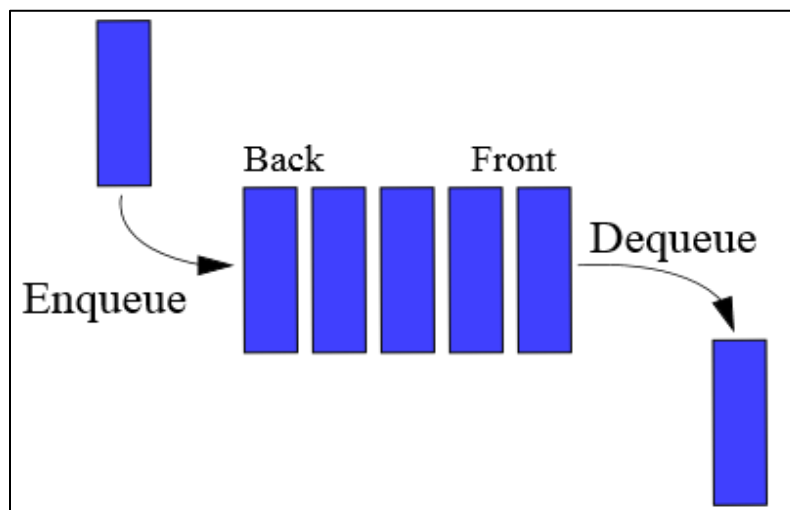


Figure 1: Represent of a Queue with FIFO (First-In-First-Out) (Wikipedia, 2013)

2.3 Queue Area

Queue area is place in which people in line, according to first-come, first-served basis, wait for goods or services. An example of a queue area would be at the bank or at the post office. Examples include checking out groceries or other goods that have been collected in a self-service shop, or in a shop without self-service, like an ATM and ticket desk. In the bank, the area usually consists of a pattern of ropes of customers forming a single line to see one of the bank officers at the counters. In another situation, customers are free to sit everywhere in

the waiting area once they have entered. Some organizations provide televisions, magazines and newspapers in the waiting area, so that customers can fill the time while waiting to be called.

Queuing can be boring, stress-filled and time consuming activity. Long waits usually make customers more likely to get upset if they feel like there has been a breach in a queue. This could happen in the event that a customer allows another customer to enter the queue in front of them instead of at the end of the queue. This is known as queue-jump. In this situation, both the person already in line in and the person attempting to enter the line need individual service. Depending on the situation, region or individuals involved, different people may allow for different breaches to queue.

2.4 QLess Mobile Wait Management

QLess is one of the virtual applications that provide queue management service to the users. The concept is easy – users are not necessary to be physically waiting in the line at the queue area, they can have their queue turn at the tips of their fingers instead. QLess provides a simple solution by alerting the users of real-time wait times at the store or office queue area, increasing demand for lower wait store or office while reducing demand for higher wait ones, thereby smoothing demand across supply.

QLess is dedicated to letting people live more and wait less. QLess Mobile Wait Management dramatically reduces waiting in line or on hold by letting people join a mobile queue from their phones, roam freely and get notified as their turn approaches. QLess also notifies hotel guests when their room is ready for check-in. QLess is the first true mobile CRM and boasts a powerful analytics & SMS surveys suite for brick-and-mortar merchants. QLess serves retail, government, education, hospitality, healthcare and everywhere people wait, and has saved millions of users well over a century of waiting.

A recent study has been conducted to look into whether not having people waiting in a waiting room for service relaxes Department of Motor Vehicles (DMVs) staff and makes them less efficient, or more. It turns out that QLess has reduced the average transaction duration at QLess DMVs throughout Kansas. The decreased stress on staff from facing happier customers rather than people tired from standing in line seems to have a dramatic improvement on productivity.

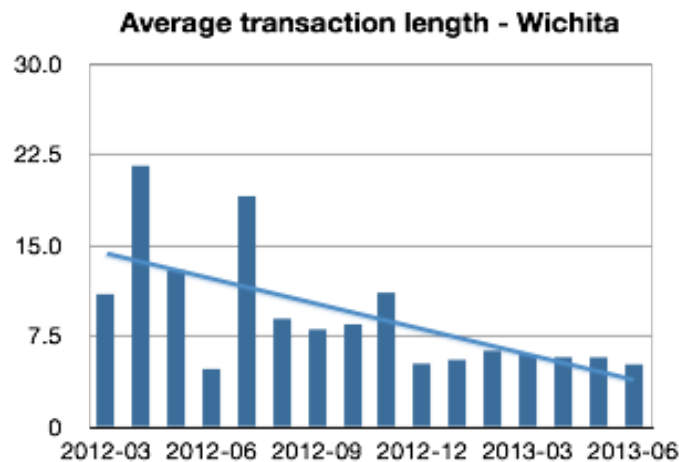


Figure 2: Average transaction length- Wichita (QLess, 2013)

2.5 Queue Algorithm

There are different data storage structures: the stack, the queue and the priority queue. The queue data structures are fundamental data structures used in this project because it stores the data in turn one by one. Robert Lafore through his book Data Structures & Algorithms in Java stated that, a queue is a data structure which the first item inserted is the first to be removed (First-In-First-Out, FIFO). According to Clifford A. Shaffer in his book Data Structures & Algorithm Analysis in Java, written that the queue is a list-like structure that provides restricted access to its element. Queue elements may only be inserted at the back, called the enqueue operation, and removed from the front, called the deque operation. While referring to Java

Programming: Program Design Including Data Structures, Chris Kiekintveld, queue is the data structure in which the elements are added at one end, called the rear, and deleted from the other end, called the front. However, the middle elements of the queue are inaccessible. Queues are used in computer's operating system for example printer queue where print jobs waits for the printer to be available and queue that stores keystrokes data as user type at the keyboard.

Based on FIFO concept, each object may only be processed according to their arrival time. While, objects are processed, others are queued and the first object should be the first to be served. There are two ways to implement the FIFO method: the array-based queue and the linked queue. The array-based queue is tricky to implement effectively because a simple conversion of the array-based list implementation is not efficient. The problems with this implementation are because of arrays having fixed sizes and after various insertion and deletion operations, the last element will be at the last array position which gives the impression that the queue is full. For the solution to the problems stated before, all the queue elements should be pushed towards the first array position. And the suggestion is to use a circular array which the front queue is defined to be toward the lower numbered positions in the array, and the rear is defined to be toward the higher numbered positions.

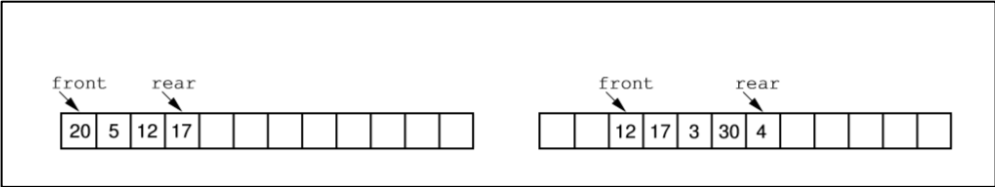


Figure 3: Array-based queue. (Java Programming: Program Design Including Data Structures)

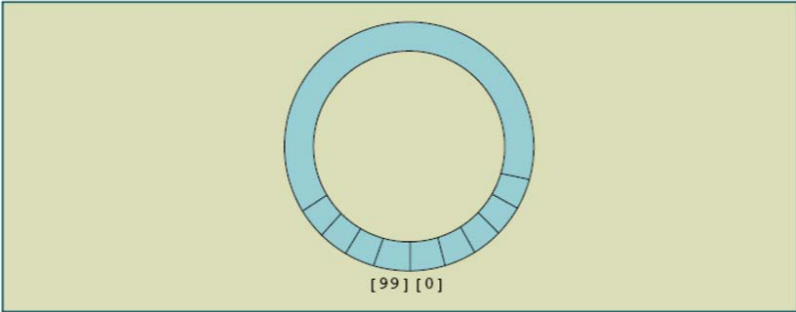


Figure 4: Circular array. (Java Programming: Program Design Including Data Structures.

On the other hand, the linked queue implementation is an exact adaptation of the linked list. It simplifies many of the special cases of the array implementation because the memory to store a queue element is allocated dynamically. In other words, the queue is never full. This implementation uses a header link node. Method enqueue places the new element in a link node at the end of the linked list and then advances rear to point to the new link node. Method dequeue grabs the first element of the list removes it.

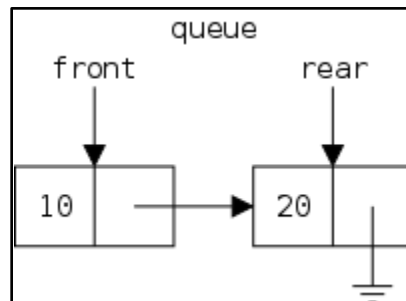


Figure 5: Linked queue. (Java Programming: Program Design Including Data Structures)

CHAPTER 3

METHODOLOGY

Research methodology refers to a set of procedures used to conduct a research project. In order to come out with a prototype, the suitable tool to create the android application of queue management system needs to be identified. The System Development Life Cycle has been adopted as the methodologies in developing the system.

3.1 System Development Methodology

Droid Queue Management was developed using Rapid Application Development (RAD) as minimal planning was required in favor of rapid prototyping. Extensive pre-planning is not necessary thus, makes it much faster to program the software and makes it easier to change the requirements in case there is any.

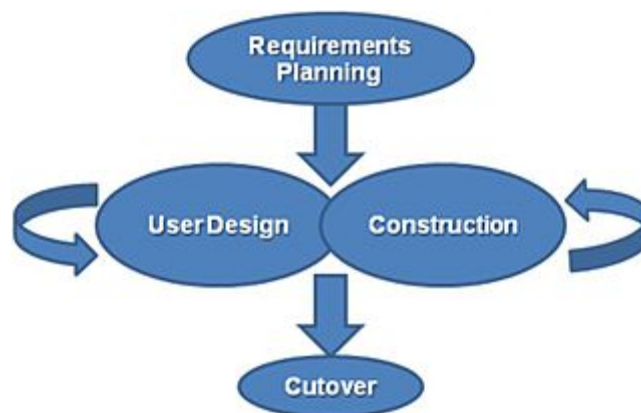


Figure 6: Rapid Application Development (RAD) Model (Toletol, 1992)

3.2 Planning

Planning phase produces a high level overview document of the project which relates to the project requirements and scopes. In this phase, the feasibility of the project was determined whether it should proceed or should not. Moreover, this phase aims to give a better understanding in terms of market analysis and technologies. Parallel with the objectives, a qualitative research is carried out as well as reviews on journal, research papers and internet sites. In order to strengthening the understanding about the application development in android environment, consultation with IT experts was done from time to time. A Gantt chart was developed to align tasks with the project time line. The purpose was to allocate sufficient time for the specific task being conducted thus ensuring that the project will be completed within the prearranged duration.

3.3 Analysis

Analysis phase determines user needs and develops user requirements. Based on the requirements determined, functional requirements of the project were discovered. During this phase, there were several steps taken into consideration in order to improve each area in this project.

- Surveys were done to gather information on market acceptance.
- User reviews on current queue management system were taken into consideration.
- Comparison between available functions on queue management system was analyzed for improvement of implementation.

3.4 Design

The design phase is the process of implementing the analysis phase into graphical user interface (GUI), conceptual design and system architecture design according to user requirements. GUI acts as a medium of communication between end users and the system application. For this purpose, a prototype designed was developed using indigo studio. Further improvement was conducted from time to time. The conceptual design is the application logic presented in form of UML diagrams. For the development of Droid Queue Management System, flow chart was constructed. System architecture design was produced based on the user requirements which involved the overview of the application as a whole. Figure 13 shows the system flow of DQMS. The red box indicates that the number request is from the customer side which is not the focus of this project.

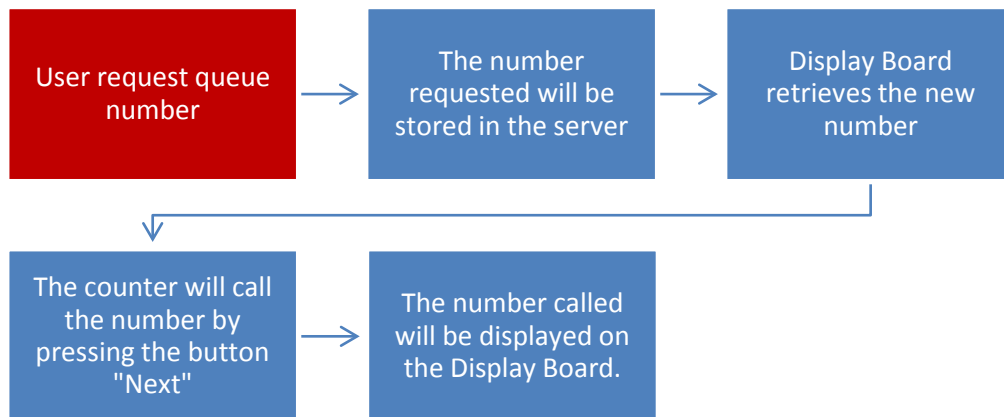


Figure 7: Flow of the DQMS

3.5 Development

The development phase is the process of implementing the analysis and the design phase into technical part which is the code. This project implements the queue data structures which also called First In First Out data structure. The approach is using array-based queue as it stores the data element in turn one by one.

However, there is a concern where adding an element to a full queue and removing an element from an empty queue would generate errors or exceptions: queue overflow exception and queue underflow exception. Therefore, the author creates classes that handle these exceptions. The classes are as below:

- QueueException extends RuntimeException
- QueueOverflowException extends QueueException
- QueueUnderFlowException extends QueueException

This implementation requires arrays to store the queue elements, thus, instance variables will be created. The variables are listed below:

- queueFront (keeps track of the first element)
- queueRear (keeps track of the last element)
- maxQueueSize (specifies the maximum size of the queue)

However, as the time goes on, there are problems determined with this implementation, and they are: arrays have fixed sizes and after various insertion and deletion operations, queueRear will point to the last array position. The solutions to those problems are to slide all of the queue elements towards the first array position and to use a circular array. In addition, the front queue is defined to be toward the lower numbered positions in the array, and the rear is defined to be toward the higher numbered positions.

This project will be using a server to store the number requested by the customers. The database was developed using PHP code in order to allow the number to be fetched by display board and the counter. This requires an internet connection for retrieving the number stored in the server. The counter will be responsible in setting the status of the queue number whether the number is a new request, currently being served or the number is done. Appendix 2 shows PHP code for the function of fetching the number from the server to the display board and counter.

3.6 Integration and test

The purpose of testing phase is to verify that the system contains all the end user requirements stated out in the analysis phase, that all the functions are accurately processing data, which the new system works with all other systems, and the new system meets the quality standards of the company and the customers.

For this project, the testing was conducted to test whether the integration between counter and display board is working or not. The integration between both prototypes is needed in order to carry out the function as the counter service in the business sectors. Furthermore, the functions developed should fulfill the user's requirement who in this case is the counter service assistant.

The test has been constructed after the system is built. The author carried out the first test of the system to 5 distinct users to perform the functions of the counter and display board respectively. The test was called pilot test because it tested all the functionality of the prototype whether it meets the requirements of the users or not. The outcomes have been recorded and there were a few feedbacks from the users that taken into consideration. The result for the pilot test is discussed further in the result and discussion chapter.

3.7 System Architecture

Figure 4 shows the system architecture for Droid Queue Management System (DQMS). However, this project is focusing on developing the queue management system for service counters. It requires Android installed tablets which have been programmed with queue management system that will allow the person in charge to call the queue number for respective service. The tablets for the service counters will control the number displayed on the display board. The counter have two options, whether to call the next number or to recall the number that have been called. And the display board will show the number called by the counter to alert the customer in the outlets. These two devices – tablets for counter and tables for display board – will be synchronized to each other in order to deliver the service in sequence. Both devices will be connected to a server which contain database of the number

requested by the customers. The server behaves as the brain of the system that connects all the devices to one database allowing the devices retrieving the numbers that had been requested.

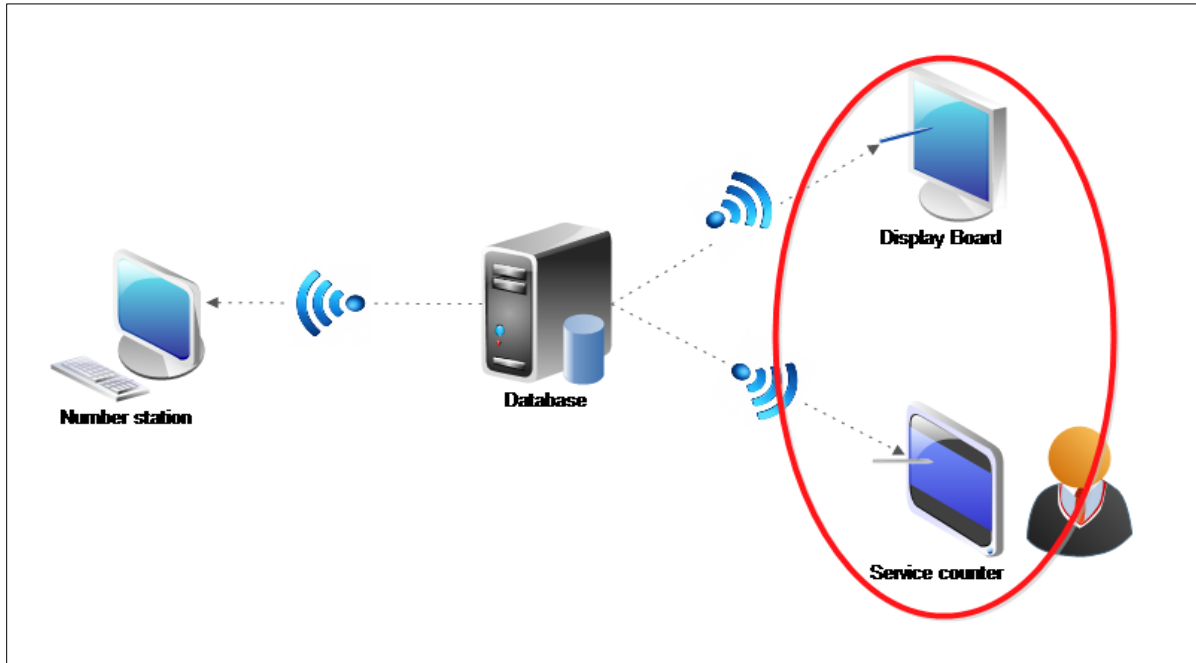


Figure 8: Droid Queue Management System Architecture

3.7 Implementation

The implementation phase is the final phase of the project where the author test and implement the completed application system before it can be delivered to the targeted audience. The completed version will be the enhanced version of the prototype based on the feedback and comment from the identified audience and users. The feedback is essential for the improvisation of the system which includes the aesthetic value and functionality requirements. This phase largely involved the technical aspect in working with codes using software such as Eclipse IDE.

3.8 Project Activities

The beginning of the project was mainly focused on research and data acquisition activities as well as understanding of the project scope. The author has undergone a familiarization of the Android environment through theoretical readings and peer discussion. Several activities have been done in making sure this project a success. A lot of investment has been made to ensure the application is practical and will bring good impact to the society.

- **Literature review**

Literature review was done in order to get a better comprehension on the current market situation for example outlets that are using queue management system. Findings from journals, research papers and internet sites signify elements to be identified and improved. Reviews were in accordance to the research element and derived results behaved as a solution contributor to the identified problem.

- **Survey and questionnaire**

Survey questions have been developed and distributed to 30 respondents. The main purpose of this survey is to support the initial hypothesis set out for this project and to validate accuracy of an intended outcome. The outcome of this survey has contributed to a holistic development of DQMS components.

- **Interview and discussion**

These activities play a very important role in each and every aspect of the development in this project. The idea of coming up with this project needs advise from experienced Android programmer to know the requirements of developing an Android application. Interview with some Android developers helped a lot in the author's motivation to continue developing the application using a handful of software at first and eventually adopt Eclipse IDE as the main environment for

software development. Besides, the interview and discussion was conducted to gather information about what needs to be included in the DQMS application in order to distinguish between DQMS and the current queue management system.

- **Android application development**

After sufficient groundwork on DQMS application, the final phase was implementation. The author has never been exposed to any Android environment apps creation finds it hard to start the ball rolling. However, the existence of online forums and online tutorials has made the learning fun and much easier to practice. After all, there are still lots of room for improvements in terms of design and functionality using the software.

Android gives the user a world-class platform for creating apps and games for Android user everywhere as well as an open marketplace for distributing them instantly. With its announcement in 2007 by Google, the purpose at hand was to devote advancing open standards for mobile devices. Building on the contributions of the open-source Linux community and more than hundreds of hardware, software and carrier partners, Android has rapidly become the fastest growing mobile operating system. (Google Inc, 2013)

Android has a powerful development framework which gives everything their user needs to build best-in-class app experiences. It gives a single application model that enable users to deploy apps broadly to hundreds of millions of users across a wide range of devices from smartphones to tablets. With new and improve devices rapidly being announce every year by phone and tablets manufactures, Android give the developer the tools for creating apps that taking the advantages of the hardware capabilities available on each devices. It will automatically adapt to the user interface of the devices and will suit each device while giving the developers ultimate control as they want over the user interface.

3.9 Key Milestone

3.9.1 Final Year Project I

Activities \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Finding Supervisor	x												
Research on propose title		x	x										
Propose final chosen title			x										
Project analysis				x	x	x	x						
Planning system design							x	x	x	x			
Project Testing										x	x		
Market Survey							x	x	x	x	x		
Analyze data												x	x

Table 1: Final Year Project I milestone

3.9.2 Final Year Project II

Activities \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Development	x	x	x	x	x	x	x	x					
Design Interface	x	x	x										
System Function				x	x	x	x	x					
System Database						x	x	x					
System Testing									x	x			
Internal Testing									x	x			
Review complete prototype with supervisor											x		
Maintenance												x	x

Table 2: Final Year Project II milestone

3.9.3 Gantt Chart

PROJECT ACTIVITIES		WEEK															
		1	2	3	4	5	6	7	Mid-Semester Break	8	9	10	11	12	13	14	
Planning	Discussion with supervisor																
	Prepare/edit Gantt Chart																
Data Collection	Conduct literature review																
	Interview with software developer & targeted user																
	Distribute survey questions																
Data Analysis	Data compilation and result identification																
	Data documentation																
System Analysis	Reviewed existing similar applications																
	Evaluate feasibility of proposed system																
Design and Development	Draft system flow and functions																
	Design user interface																
	Develop prototype																
	System testing																
	Evaluate system																
Documentation	Complete report																
Important Dates	Progress report				*												
	Pre-EDX																
	Dissertation																
	VIVA																
	Technical report																

The symbol * indicates key milestone for this project.

3.10 Tools

3.10.1 Software

- Eclipse IDE
- Android Software Development Kit (SDK)
- EDraw Max
- Indigo studio
- Microsoft Office

3.10.2 Hardware

- ASUS A53T
- Android Tablet

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

Data gathering is essential for supporting the previously identified hypothesis. It validates the purpose of the project and helps in the holistic development of the application. For gathering data, a questionnaire survey was conducted. The survey comprises of 8 sets of question and divided into 4 parts: client's aspect, current system's issues, acceptance of Android application and inclination towards green IT products. The survey was then distributed to 30 respondents working for different banks in Batu Gajah. Result from the survey will be discussed more in next section.

4.2 Data Gathering

4.2.1 Survey Analysis

- **Survey Objectives**

The purpose of the survey/questionnaire is to analyze demographical preference of the target audience towards using the DQMS and familiarization with Android application.

- **Data Collection Process**

The survey was conducted on 30 respondents of targeted users working in different banks in Batu Gajah includes CIMB Bank, Maybank, Hong Leong Bank and Bank

Simpanan Nasional . The reason of choosing banks as target market is because they are using the current Queue Management System in their offices to serve clients in everyday business. The survey was conducted manually on paper in order to interact directly with the respondents and to observe the real situation in the bank outlets.

- **Findings**

Each questionnaire was divided into four parts: client’s aspect, current system’s issues, acceptance of Android application and inclination towards green IT products. 30 respondents had participated actively in the survey and findings will be further discussed in the following sections of this report.

Part 1: Client’s Aspect

This part of the survey behaves as the preliminary stage and focuses on client aspect which is the main reason to implement DQMS. Aspect of the number of clients served per day as well as smartphone ownership was taken into account.

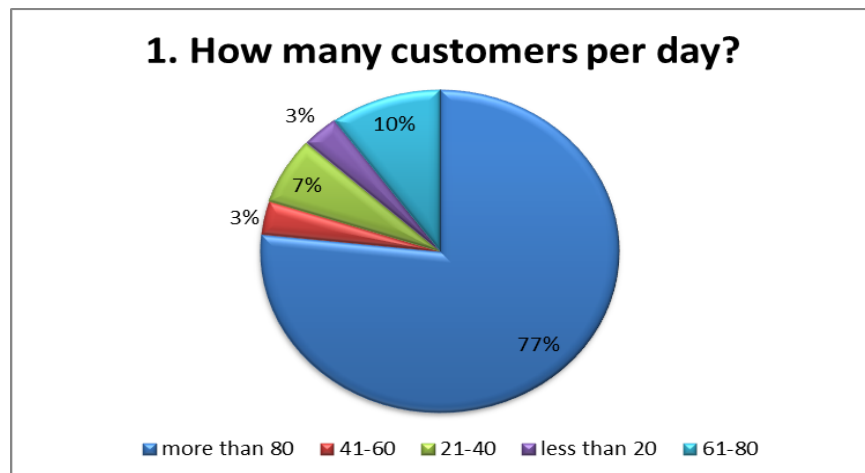


Figure 9: Number of clients per day

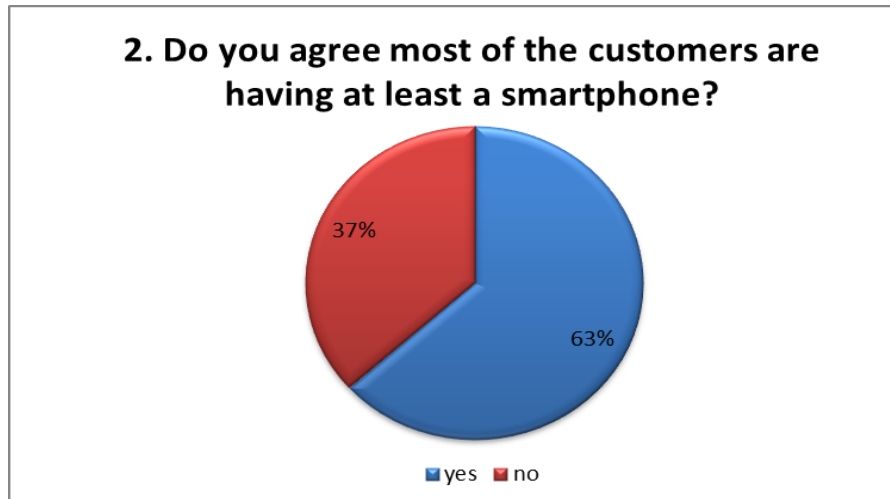


Figure 10: Smartphone ownership

This part shows that 23 (77%) out of 30 respondents served more than 80 clients per day, and 19 (63%) out of 30 respondents agreed that most of the clients are having at least a smartphone. Some of the respondents pointed out that because of technology evolution nowadays most of the clients are encouraged to have smartphones to help them in almost all their daily activities. One important information that the author tried to grasp in this section was whether or not the clients own a smartphone which in this case is very significant because DQMS can only be integrated with a smartphone.

Part 2: Current System Issues

This part of the survey is aimed at identifying the respondents' behaviors towards current queue management system and issues that they faced. Both of the questions are related and only respondents who agreed in question 3 have to answer question 4, an open ended question, stating the problems that they faced with the current queue management system.

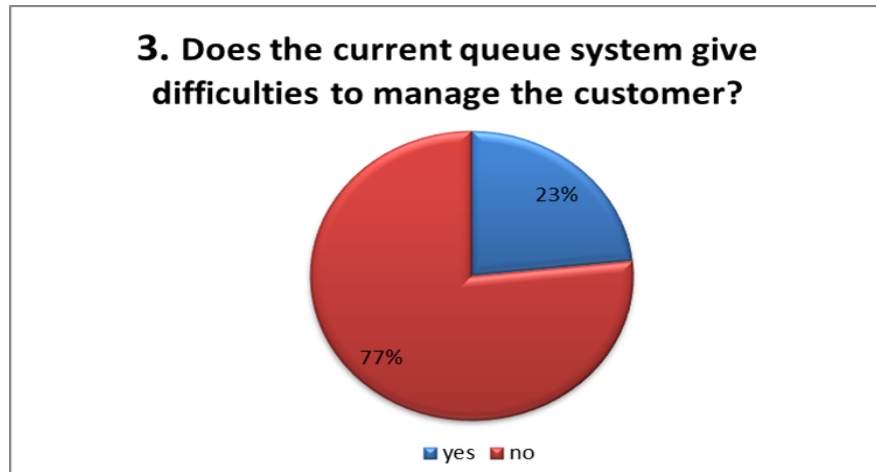


Figure 11: Current Queue Management

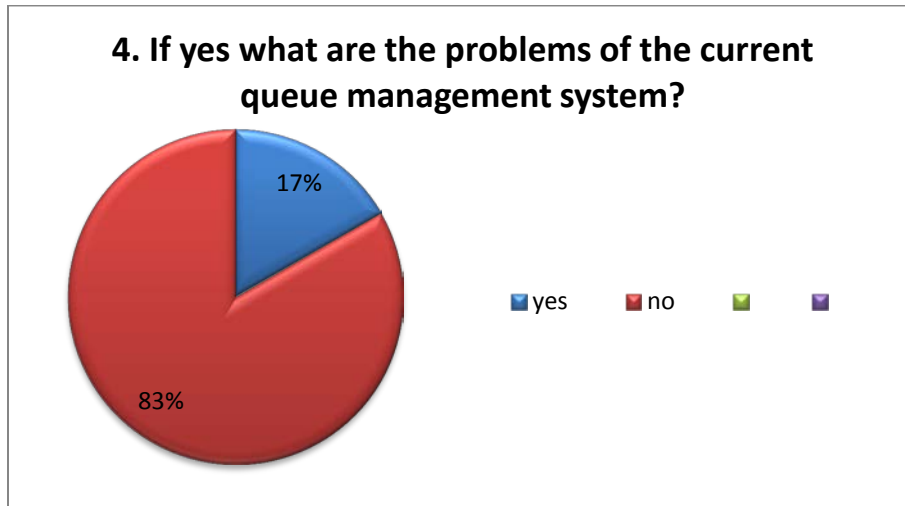


Figure 12: Current Queue Management issues

This part shows only a small portion of the respondents agreed that the current queue management system give difficulties to manage the clients which is 7 (23%) out 30 respondents. Among the 7 respondents, 3 of them stated that, they have to face clients inappropriate behaviors for instance some of the clients do not wait until their turn instead they rush to the counters and ask to be served because they have something else to be done. Other respondents stated that time consuming (29%), parking availability (14%) and different service with different number (14%).

Part 3: Acceptance of Android Application

This part of the survey portrays the connection between developments of the application to the user's acceptance towards it. This section discovers two main factors in the decision of users to adopt a technology (mobile application).

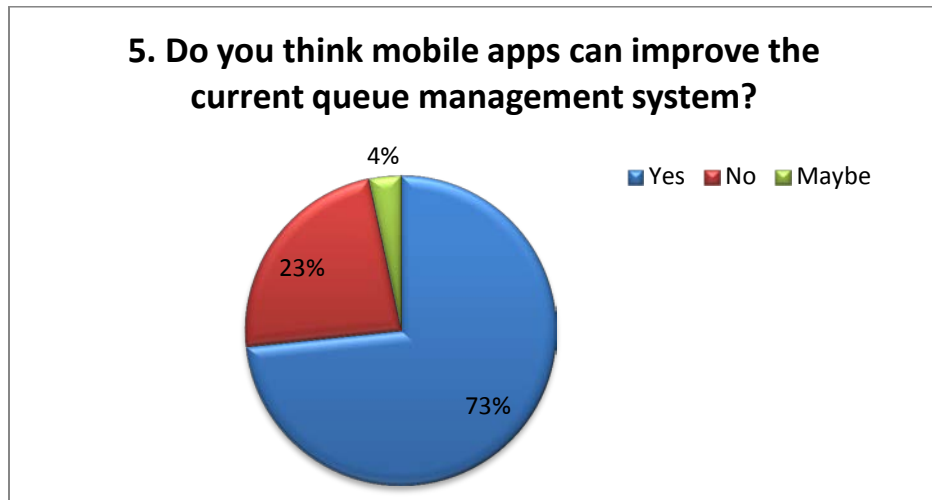


Figure 13: Improvement on queue management system

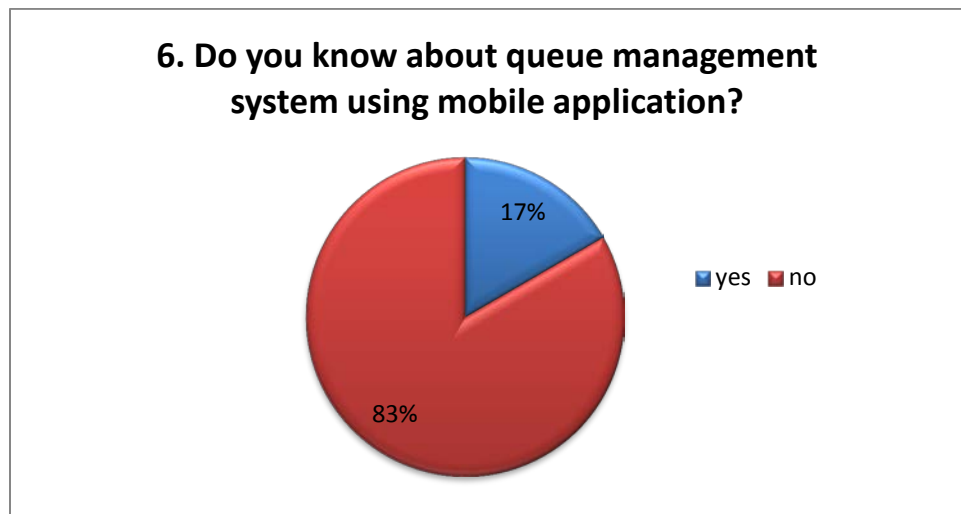


Figure 14: Familiarity of mobile apps in QMS

This part shows that 22 (73%) out of 30 agreed that mobile apps has a high possibility to improve the current queue management system. While only 5 (17%) out of 30 respondents know about queue management system in mobile application.

Part 4: Inclination towards green IT products

This section is aimed to validate the importance of the green IT product, like the proposed application. Opinions and perceptions from the respondents are collected to ensure the application has a valid functionality and serve its purpose.

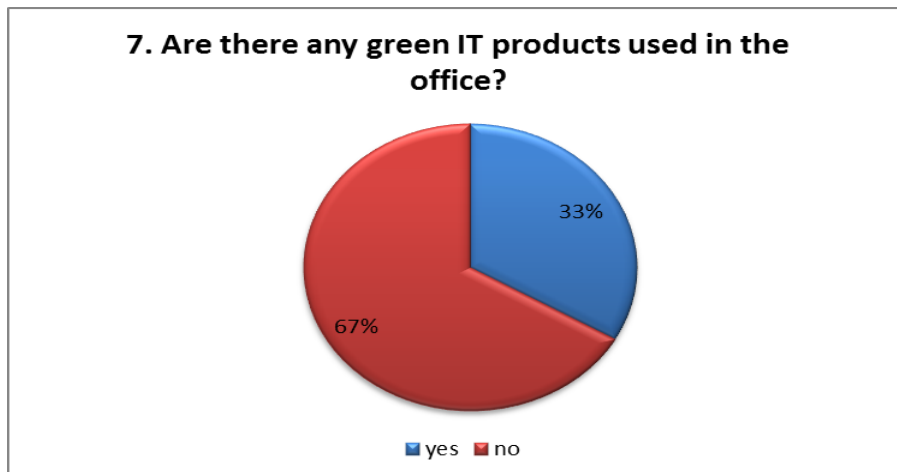


Figure 15: Familiarity of green IT usage

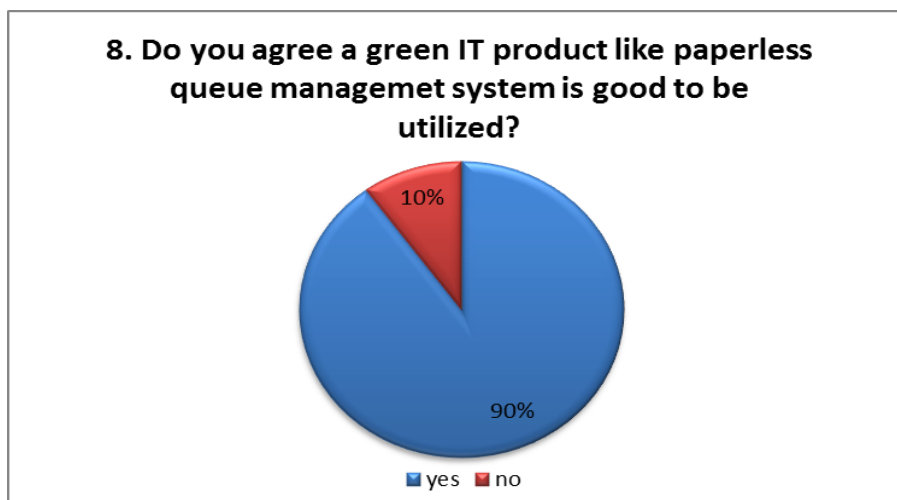


Figure 16: The importance of green IT products

This part shows 10 (33%) out of 30 respondents are using green IT product in the offices and are aware about “Go green” campaign. The last question reflects the acceptance of the green IT product by the respondents and 27 of them agreed that green IT product is good to be utilized.

4.3 Prototyping

In the prototyping cycles, there will be two main phases which are user design and construction. After the user design phase has been finalized, the system prototype will be build. As been planned in the key milestone, the prototype development will be started in Final Year Project II. However, the initial design of the system interfaces will be done in Final Year Project I. Next section will discuss about the initial designs that have been done prior to the stage of developing prototype.

The development of this system will start on the development phase in Final Year Project II. From the key milestone, the development process will take eight weeks to complete. There will be three activities in the development process. The first activity is to design the real interface for the system. The interfaces design will be based on the initial design that has been done in Final Year Project I. The real interfaces design process is expected to finish in three weeks duration which includes the improvisation of the design parallel to system requirements.

After the interfaces have been completed, the author will start to develop the system function. The development of system function will take five weeks to complete. The author will use Android SDK development tools with the Google Chart API libraries. All the programming code is written and compile by using an open source software Eclipse. For the DQMS database, developer will use SQLite Android. Basically, there is just a plenty of data which is the queue number. The queue number data will be stored temporarily because it needs to be reset and restored for another system testing.

In the demonstrate process, developer will use android emulator to test run the system prototype. The aim of this process is to evaluate the system prototype and to ensure the deliverable of the system. If the system prototype is completely functional and meets the entire system requirement, then the prototype is ready to proceed with system testing stages. However, if the system prototype is fail during the demonstration process, the prototype will undergo the refining process.

In the refining process, the author will re-build the system prototype and fix the bugs found in the previous system prototype. The prototyping cycles will be continued until the complete working prototype is done and ready to the next stages.

4.4 Pilot test result

Pilot testing is a process of testing all the functionality of the prototype. It tests whether the prototype meets the requirement of the user. Basically, it means to check whether the functionalities are working as expected or not. It helps in making decision on the system readiness for production after taking a few considerations such as what are the errors found, did the system work as expected, what are the adjustment needed etc.

For this project, pilot testing is conducted after the system has been developed and ready to be tested in the user-like environment. Five users have been selected to try the prototype as if they were counter service assistants. First of all, they were explained on how is the current queue management system works and they have been emphasized to that the current system is using paper ticket number. Then, they were explained on how DQMS works which is paperless based on the developed system. Even though this project is focusing on the counter service side, the author has developed a prototype to generate the queue number in order to store the number in the server and allow the number to be fetched by the display board and the counter. As for now, the author assumed that it will be one counter delivers one service to the customers.

In order to test the functionality of the prototype, the test is conducted exactly like the system flow. Even though, the customer side is out of focus of this project, it is done to make sure that the number requested is able to be fetched by the counter service side.

1. The user requested queue number by using the generating number prototype that has been developed by author. The prototype will send the number requested by the user and store the number in the server. Figure 14 shows that a number is requested by the user.



Figure 17: Number requested by the user.

2. The user then was shown that the number requested is displayed on the display board in the pending column. Display board interface has two boxes – one is for processing number and the other one is for pending number. Figure 15 shows that number 65 and 66 is already got in the pending box. It means that the number has not been called by the counter yet.

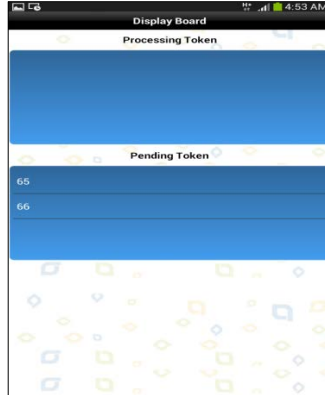


Figure 18: Display board shows the pending number.

3. Then the user acted as a counter service assistant and the user was asked to process the number 65. Figure 16 shows the counter interface which has two buttons. The first button is “Accept” and the second button is “Done”.

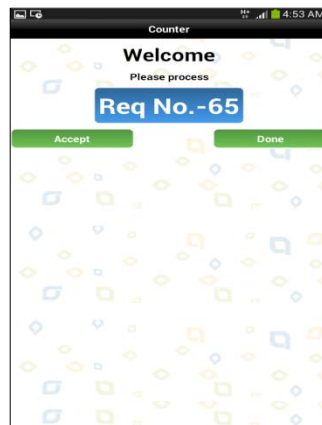


Figure 19: Counter prompted the user to process the number 65.

4. The user tapped on the “Accept” button to process the number. Figure 17 shows that the number 65 is currently processing after the user tapped on the “Accept” button. It means that the number 65 is receiving the service.

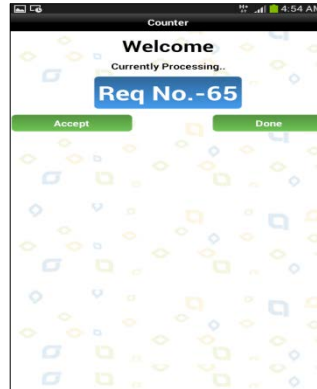


Figure 20: Counter shown that the number is being processed.

5. While the number is in the processing state, the display board will show the currently processing number in the processing box. Figure 18 shows that the number 65 is got in the processing box. It means that the number 65 has been called by the counter and is currently being served.

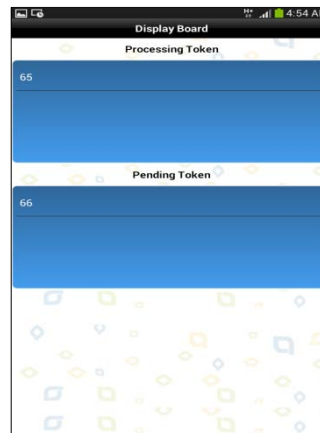


Figure 21: Display board shown the number 65 is currently being served.

6. The user was aware about the changes happened is synchronized between the counter and the display board. For the last step of the test, the user tapped on the “Done” button as if the service is finished. As the “Done” button was tapped, the counter will show the next number which is 66 and asked the user to process the number. Figure 19 shows the counter prompted the next number to be process.

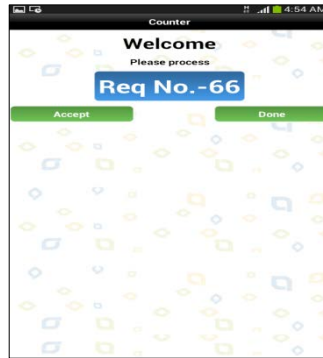


Figure 22: Counter prompted the next number to be processed.

Basically, this test was successfully conducted and helps the users to comprehend the function of the DQMS and how it works. They understood that the counter device and the display board device were related and synchronized to each other in order to perform the system. However, after the test was done, there were few feedbacks that should have been taken into author's consideration. The feedbacks are as follows:

1. There should be a repeat button for the counter to recall the number if the customer does not notice his/her number is being called. This function will be synchronized with the display board to blink the recalled number.
2. The counter should be able to deliver different services thus requires different range of queue numbers based on the service available. It must be tested on few numbers of counters so that the system will clearly replicate the real situation of delivering service over the counter.
3. The counter should have the login page to activate the system so that the number will be distributed properly according to the service provided.

4.4.1 Counter service side functionalities

The test conducted was emphasized on the functionality of the counter prototype and also the display board prototype, since both of these devices is used by almost all the outlets which have counter service. Below are the summaries of the functionalities for both prototypes.

- Counter

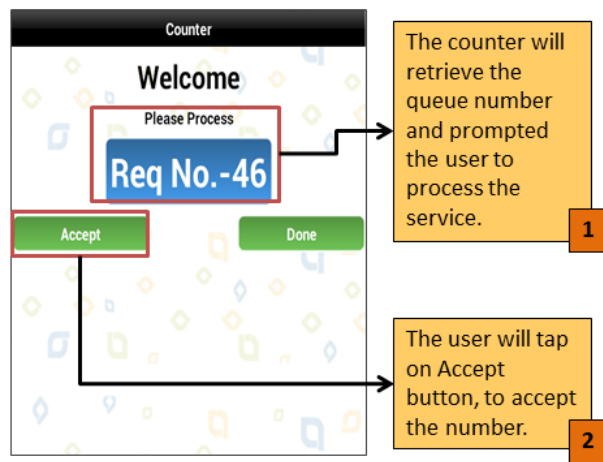


Figure 23: Counter retrieves queue number.

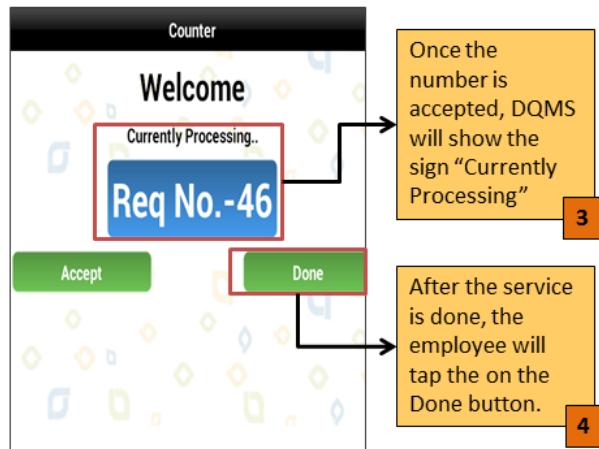


Figure 24: The user accepts the number.

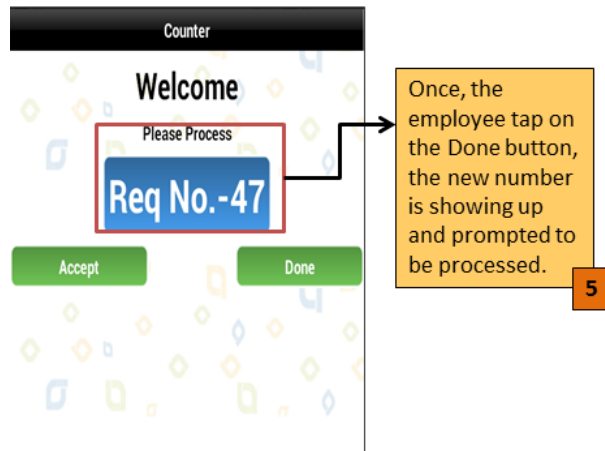


Figure 25: The service is done.

- Display Board

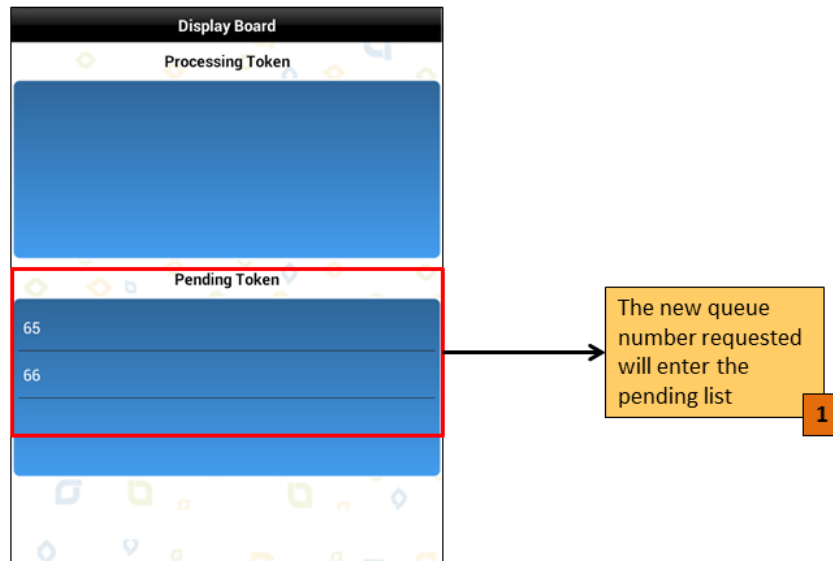


Figure 26: Displays the number that hasn't received the service.

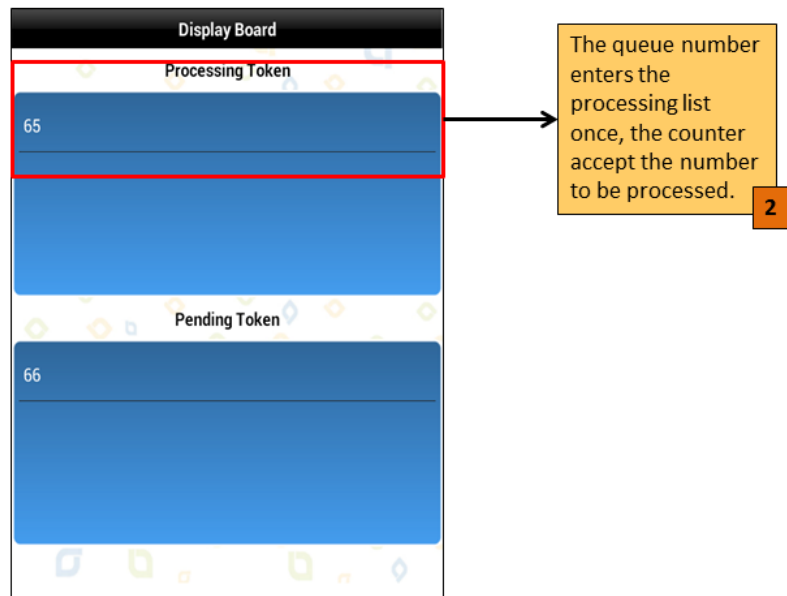


Figure 27: Displays the number that being served.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Generally, the development process of DQMS application was parallel to the objectives previously identified. Firstly, research on current queue management system in order to get a better understanding on fundamental requirements to incorporate Android application as a mean to provide service to clients. For planning phase, sufficient research was conducted to provide a solid platform for project initiation. Subsequently, the analysis phase was primarily focused on the pre-implementation survey and DQMS application market research findings. The survey outcome was successful in validating the feasibility and identifying system requirements of DQMS application.

The development of system will be using the Rapid Application Development (RAD) method. There will be four main phases in the RAD which are Requirement analysis and system design, prototyping cycles, system testing and implementation. The system will be developing using Android SDK development tool, an application development tool and also use Google Chart API libraries.

By taking the initiative of Green Computing technology, this system could help promote the “Go Green” campaign by creating an awareness to improve paper waste management thus preventing greenhouse effect which can harm the environment. In align with government encouragement in using smartphones, this application has a better future in the market as it helps society to manage time for daily activities.

5.2 Future work

For improvement purposes, several suggestions can be imposed on Droid Queue Management System based on the conducted pilot test. The interface of the counter should add “Repeat” button to recall the number just in case if the customer is not aware of the number call. This means whenever the counter service assistant taps on the “Repeat” button, the number on the display board will blink to alert the customer.

In order to replicate the real situation faced by the counter service assistants, the queue number should be divided into few ranges for different services. For example, most of the banks have different services such as investment, loan and saving etc. Therefore, for each service, there should be different ranges of queue number so that the service will be delivered much efficient to the customer.

Furthermore, the counter should have a login page to activate the system. The purpose of having a login page is to make sure that each counter is active for the service it provides. This will allow the queue numbers to be distributed only to the active counters, so that the customers will not mistakenly go to the wrong counter to settle their business. This will ensure an organized situation in the targeted outlets.

In addition, the application is also suggested to improve the connection speed between all DQMS devices which include the connection between DQMS application with DQMS controller, DQMS application with wireless printer, and DQMS application with the internet by putting a local server in the office area. By implement a local server, DQMS can improve their efficiency and effectiveness of its operation with improved speed of operation without any interruption if the internet is down.

Perhaps, DQMS can also be replicated into multiple Operating Systems (OS) like iOS and Symbian, just to name a few. The market potential of Smartphone is now huge and being able to leverage on its potential would surely benefited a lot to the industry.

REFERENCES

- [1] Abd Hamid, Muhammad Ikmal. *Droid Queue Management System* (2012): Print.
- [2] C.A. Shaffer, *A Practical Introduction to data Structures and Algorithm Analysis: Second Edition*, Prentice Hall, Upper Saddle River, NJ, 2001. ISBN 0-13-028446-7
- [3] Froutan, P. (2004) "How to defend against DDoS attacks" Computerworld Magazine, June 24, 2004. Retrieved October 28, 2011.
- [4] G. Kornaros, I. Papaefstathiou, A. Nikologiannis, N. Zervos, *A Fully-Programmable Memory Management System Optimizing Queue Handling at Multi Gigabit Rates*, dac, pp.54, 40th Design Automation Conference (DAC'03), 2003
- [5] Gosha, Kinnis K. *Queueadmin: The effects of an advance queue management system on barbershop administration* (2007): http://etd.auburn.edu/etd/bitstream/handle/10415/74/GOSHA_KINNIS_25.pdf . July 29, 2013
- [6] Höpfner, H., Pencke, J., Wiesner, D., and Schirmer, M. (2011). Towards a Target Platform Independent Specification and Generation of Information System Apps. ACM SIGSOFT Software Engineering Notes, 36(4), 1-5.
- [7] Kiekintveld C. (2010), *Java Programming: Program Design Including Data Structures*. http://www.cs.utep.edu/vladik/cs2401.10a/Ch_14_Recursion.pdf . July 29, 2013
- [8] Queue (abstract data type). *Wikipedia.com*. Retrieved July 22, 2013, from [http://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](http://en.wikipedia.org/wiki/Queue_(abstract_data_type))
- [9] Qless. *Qless.com*. Retrieved July 29, 2013, from www.qless.com

[10] Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2003). *Systems Analysis and Design Methods*. 6th edition. ISBN 0-256-19906-X.

[11] Yuval, F., Uri, K., Yuval, E., Shlomi, D., Chanan (2010). “Google Android : A Comprehensive Security Assessment”. *IEEE Security & Privacy* (IEEE) (in press). D-O- I : 10.1109/MSP.2010.2.

APPENDICES

Appendix 1: Pre-Implementation Survey

Name:

Company:

The Droid Queue Management System (DQMS) is an android application designed to promote green computing technology by providing a paperless queue ticketing service. The main purpose of this project is to provide an alternative way to the current queue management system which uses paper ticket. This application will provide a queue number in form of digital ticket direct to user's smartphone through the QR code generate by a QR code generator device.

1. How many customers per day?
 - a. Less than 20
 - b. 21-40
 - c. 41-60
 - d. 61-80
 - e. More than 80

2. Do you agree most of the customers are having at least a smartphone?
 - a. Yes
 - b. No

3. Does the current queue system give difficulties to manage the customers?
 - a. Yes
 - b. No

4. If yes, what are the problems of the current queue system?
State: _____

5. Do you think mobile apps can improve the current queue management system?
 - a. Yes
 - b. No

6. Do you know about Queue Management System using android application?
 - a. Yes
 - b. No

7. Are there any green IT products used in the office?
 - a. Yes
 - b. No

8. Do you agree a green IT product like paperless queue management system is good to be utilized?
 - a. Yes
 - b. No

Appendix 2: PHP Code

1. getallrequest

```
<?php
require_once("db.php");

$sql = "select * from tokens order by tokenid";
if ($_REQUEST['rows']) {
    $sql .= " limit {"$_REQUEST['rows']}";
}
$result = mysql_query($sql, $myConn);

$json = array();

while ($row = mysql_fetch_assoc($result)) {
    $json[] = $row;
}

echo json_encode($json);
?>
```

2. getcurrentrequest

```
<?php
require_once("db.php");

$sql = "select * from tokens where status = 'current' order by
tokenid";
$result = mysql_query($sql, $myConn);

$json = array();

while ($row = mysql_fetch_assoc($result)) {
    $json[] = $row;
}

echo json_encode($json);
?>
```

3. getrequest

```
<?php
require_once("db.php");

$sql = "select * from tokens where status = 'new' order by tokenid";
if ($_REQUEST['rows']) {
    $sql .= " limit {"$_REQUEST['rows']}";
}
$result = mysql_query($sql, $myConn);

$json = array();

while ($row = mysql_fetch_assoc($result)) {
    $json[] = $row;
}

echo json_encode($json);
?>
```

4. newrequest

```
<?php
require_once("db.php");

$requestid = $_REQUEST['requestid'];

if (!$requestid) {
    $requestid = md5(date("yyyy-mm-dd:H:i:s:nn"));
}

$sql = "insert into tokens(requestid) values('{$_requestid}')";
mysql_query($sql, $myConn);

$sql = "select * from tokens where requestid = '{$_requestid}'";
$result = mysql_query($sql);
$row = mysql_fetch_assoc($result);

echo json_encode($row);
?>
```

5. setstatus

```
<?php
require_once("db.php");

$tokenid = $_REQUEST['tokenid'];
$status = $_REQUEST['status'];

$sql = "update tokens set status = '{$status}' where tokenid =
'{$tokenid}'";
mysql_query($sql, $myConn);
?>
```