**Queuing Management System with SMS Notification and Prioritization**

**A Capstone Project**

**presented to the**

**Department of Information Technology**

**Faculty of Engineering**

**University of Santo Tomas**

**in partial fulfilment**

**of the requirements in the degree of**

**Bachelor of Science in Information Technology**

**Saguinsin, Patrick Joshua B.**

**Santos, Maidy Precious Paula E.**

**Diza, Justine Rianne R.**

**Urzabia, Jasmine Eve T.**

**Santos, Anne Michelle**

**October 2014**

Table of Contents

[Chapter 1 Introduction 1](#_Toc401707369)

[1.1 Project Context 1](#_Toc401707370)

[1.2 Purpose and Description 2](#_Toc401707371)

[1.3 Objectives 3](#_Toc401707372)

[1.4. Scope and Limitations 3](#_Toc401707373)

[Chapter 2 Review of Related Literature, Studies, and Systems 5](#_Toc401707374)

[Related Literatures and Studies 5](#_Toc401707375)

[Related Technologies 12](#_Toc401707376)

[Chapter 3 Technical Background 15](#_Toc401707377)

[3.1 Design Methodology 15](#_Toc401707378)

[3.2 Requirements Analysis 16](#_Toc401707379)

[3.2.1 Functional Requirements: 16](#_Toc401707380)

[3.2.2 Non-Functional Requirements: 17](#_Toc401707381)

[3.3 Overview 17](#_Toc401707382)

[3.3.1 System requirements 20](#_Toc401707383)

[3.3.2 Prototype 22](#_Toc401707384)

[Chapter 4 Implementation, Results and Discussion 31](#_Toc401707385)

[4.1 Requirements Documentation 31](#_Toc401707386)

[4.3 Development and Testing 45](#_Toc401707387)

[4.4 Description of the Prototype 62](#_Toc401707388)

[Chapter 5 Recommendations 73](#_Toc401707389)

[Bibliography 75](#_Toc401707390)

**Table of Figures**

[Figure 1.1 – Block Diagram of the Queuing System 2](#_Toc401708488)

[Figure 3.1.1: Agile Development Model 15](#_Toc401708489)

[Figure 3.3.1: Basic system operation 18](#_Toc401708490)

[Figure 3.3.2: Context Diagram (Data Flow Diagram - Level 0) 19](#_Toc401708491)

[Figure 3.3.3: Flowchart of the system 20](#_Toc401708492)

[Figure 3.3.2.1: System Architecture 23](#_Toc401708493)

[Figure 3.3.2.2: Home Screen for desktop 25](#_Toc401708494)

[Figure 3.3.2.3: Home screen for mobile 26](#_Toc401708495)

[Figure 3.3.2.4: VIP screen for desktop 27](#_Toc401708496)

[Figure 3.3.2.5: VIP screen for mobile 27](#_Toc401708497)

[Figure 3.3.2.5: Real-time screen for desktop 28](#_Toc401708498)

[Figure 3.3.2.7: SMS Confirmation & Notification received by the Client 29](#_Toc401708499)

[Figure 3.3.2.8: Controller interface 30](#_Toc401708500)

[Figure 3.3.2.9: Updating or modifying transaction list 30](#_Toc401708501)

[Figure 3.3.2.10: Adding a new transaction 31](#_Toc401708502)

[Figure 3.3.2.11: Display screen 31](#_Toc401708503)

[Figure 4.1.2: Client use case diagram 34](#_Toc401708504)

[Figure 4.2.2.1: Home page sequence diagram 39](#_Toc401708505)

[Figure 4.2.2.2: VIP page sequence diagram 40](#_Toc401708506)

[Figure 4.2.2.3: Receiving SMS notification sequence diagram 41](#_Toc401708507)

[Figure 4.2.2.4: Real-time page sequence diagram 42](#_Toc401708508)

[Figure 4.2.2.5: Controller sequence diagram 43](#_Toc401708509)

[Figure 4.2.3.1: Network Interconnection 44](#_Toc401708510)

[Figure 4.2.4.1: Database Structure 45](#_Toc401708511)

[Figure 4.4.2: Web page that display the ticket number after requesting 65](#_Toc401708512)

[Figure 4.4.3: VIP Login page 66](#_Toc401708513)

[Figure 4.4.4: Web page after login 66](#_Toc401708514)

[Figure 4.4.1: Real-time Queue Status Page 67](#_Toc401708515)

[Figure 4.4.6: Administrator Module Security Feature 68](#_Toc401708516)

[Figure 4.4.8: Controller 70](#_Toc401708517)

[Figure 4.4.8.1: Connecting to System Database 71](#_Toc401708518)

[Figure 4.4.8.2: Database Display 72](#_Toc401708519)

[Figure 4.4.9: Display 73](#_Toc401708520)

**List of Tables**

[Table 4.3.1 Home page (desktop view) – correct cellphone number with SMS notifications enabled 46](#_Toc401710415)

[Table 4.3.2 Home page (desktop view) – correct cellphone number with SMS notifications disabled 46](#_Toc401710416)

[Table 4.3.3 Home page (desktop view) – correct cellphone number with SMS notifications enabled then cancelled request 47](#_Toc401710417)

[Table 4.3.4 Home page (desktop view) – correct cellphone number with SMS notifications disabled then cancelled request 47](#_Toc401710418)

[Table 4.3.5 Home page (desktop view) – incorrect cellphone number with SMS notifications enabled 48](#_Toc401710419)

[Table 4.3.6 Home page (desktop view) – incorrect cellphone number with SMS notifications disabled 48](#_Toc401710420)

[Table 4.3.7 VIP login (desktop view) – login with correct password with SMS notifications and prioritize me enabled 49](#_Toc401710421)

[Table 4.3.8 VIP login (desktop view) – login with correct password with SMS notifications disabled 49](#_Toc401710422)

[Table 4.3.9 VIP login (desktop view) – login with incorrect password 50](#_Toc401710423)

[Table 4.3.10 Real-time queue (desktop view) 50](#_Toc401710424)

[Table 4.3.11 Home page (mobile view) – correct cellphone number with SMS notifications enabled 50](#_Toc401710425)

[Table 4.3.12 Home page (mobile view) – correct cellphone number with SMS notifications disabled 51](#_Toc401710426)

[Table 4.3.13 Home page (mobile view) – correct cellphone number with SMS notifications enabled then cancelled request 51](#_Toc401710427)

[Table 4.3.14 Home page (mobile view) – correct cellphone number with SMS notifications disabled then cancelled request 52](#_Toc401710428)

[Table 4.3.15 Home page (mobile view) – incorrect cellphone number with SMS notifications enabled 52](#_Toc401710429)

[Table 4.3.16 Home page (mobile view) – incorrect cellphone number with SMS notifications disabled 53](#_Toc401710430)

[Table 4.3.17 VIP login (mobile view) – login with correct password with SMS notifications and prioritize me enabled 53](#_Toc401710431)

[Table 4.3.18 VIP login (mobile view) – login with correct password with SMS notifications disabled 54](#_Toc401710432)

[Table 4.3.19 VIP login (mobile view) – login with incorrect password 54](#_Toc401710433)

[Table 4.3.20 Real-time queue (mobile view) 54](#_Toc401710434)

[Table 4.3.21 Controller (with active queue) – view queue table 54](#_Toc401710435)

[Table 4.3.22 Controller (with active queue) – view records history 55](#_Toc401710436)

[Table 4.3.23 Controller (with active queue) – view VIP list 55](#_Toc401710437)

[Table 4.3.24 Controller (with active queue) – add a VIP with correct number format and confirmation password 56](#_Toc401710438)

[Table 4.3.25 Controller (with active queue) – add a VIP with incorrect mobile number and correct confirmation password 56](#_Toc401710439)

[Table 4.3.26 Controller (with active queue) – add a VIP with correct mobile number and incorrect confirmation password 57](#_Toc401710440)

[Table 4.3.27 Controller (with active queue) – add a VIP with incorrect mobile number and incorrect confirmation password 57](#_Toc401710441)

[Table 4.3.28 Controller (with active queue) – add a transaction 58](#_Toc401710442)

[Table 4.3.29 Controller (with active queue) – edit VIP record 59](#_Toc401710443)

[Table 4.3.30 Controller (with active queue) – delete VIP record 59](#_Toc401710444)

[Table 4.3.31 Controller (with active queue) – edit transaction 59](#_Toc401710445)

[Table 4.3.32 Controller (with active queue) – delete transaction 60](#_Toc401710446)

[Table 4.3.33 Controller (with active queue) – cleanup queue 60](#_Toc401710447)

[Table 4.3.34 Controller (with active queue) – clear queue 61](#_Toc401710448)

[Table 4.3.35 Controller (with active queue) – clear records history 61](#_Toc401710449)

[Table 4.3.36 Controller (with active queue) – launch display 62](#_Toc401710450)

[Table 4.3.37 Controller (with active queue) – fullscreen display 62](#_Toc401710451)

[Table 4.3.38 Controller (with active queue) – restore window controls 62](#_Toc401710452)

[Table 4.3.39 Controller (with active queue) – close display 62](#_Toc401710453)

[Table 4.3.40 Controller (with active queue) – call next 63](#_Toc401710454)

[Table 4.3.41 Controller (with active queue) – call again 63](#_Toc401710455)

# Chapter 1 Introduction

## 1.1 Project Context

Back in the day, when people would opt to execute some business transactions, most often you would find them falling in line. Whether they end up standing or sitting while waiting for their turn, this brings about a lengthy period of waiting resulting in low customer satisfaction and poor company image - based on a research conducted by Barclay and his fellow colleagues (2013), results showed that psychological morbidity worsened during the period of waiting also,David Maister in his theory Psychology of Waiting Lines stated that waiting-line experience in a service facility significantly affects our overall perceptions of the quality of service provided. These days, given the immensity of actions that need be done, simple queuing systems have been the stagnant choice for handling lines of people for transactions in an organization - usually in the service-providing businesses.

Current queuing systems handle lines in a more effective way by providing a number and allowing the clients to wait. However, people must be physically present - which still brings the old issue of inconvenience to those who wish to do more productive acts for their time spent waiting instead of staying in the queue. This brought about a solution for such a problem, instead of merely providing a number and letting the clients wait, an additional feature was implemented - a web application incorporated with SMS notifications. Now, people can spend their time productively while waiting for their turn in the queue. Also, another feature was added - Prioritization. Those who have certain privileges have instant access to the front of the queue, providing them with convenience and a higher rate of satisfaction. Not only does this make it more convenient for the client, the administrator of the transactions also have ease in completing the transaction since the purpose of their stay is indicated in the queue request. No more questions, no more indefinite waiting, convenience and ease are served.

* Phone Number
* Username and password of VIP clients
* Queue Number
* Queue Status
* Priority
* Priority Status
* Gives client a notification of their status in the queue
* Gives priority to VIP clients

## 

## 

## 

## 

Figure 1.1 – Block Diagram of the Queuing System

## 1.2 Purpose and Description

The project aims to alleviate the problems existing in current typical queueing systems by adding features that will effectively solve time-consuming problems by creating a queuing management system prototype which will initially run on the team’s personal devices. The prototype integrates SMS Notifications and Prioritization. Target beneficiaries include the clients and customers of an organization or a business entity. The project shall use the combination of general purpose computer systems to balance the required power over cost ratio requirements of each unit in the system. In addition, a Short Message Service Center (SMSC) will be used to accommodate the SMS feature of the new queueing system.

This is all in an effort to discard the dilapidated system of handling transactions - giving clients timeliness, to automate the action of ‘falling in line’ - allowing the clients to roam freely while waiting for their turn; getting rid of the stress and decreasing client dissatisfaction.

## 1.3 Objectives

The proposed project aims to implement queuing concepts meshed with two additional features - namely SMS Notifications and Prioritization - to provide practical benefits for the business' clients. Goals to meet the project expectations include: sending of SMS notifications to clients (who have availed for this type of service) of their current queue status, prioritizing VIP guests, easing transaction handling in the case of the business administrator (ex. bank teller), minimizing the use of paper, and allowing the clients to spend their time productively while waiting for their turn.

## 1.4. Scope and Limitations

The system routine shall support basic functions and processes of receiving input from a client. The request shall be placed on the queue, depending on the priority status, whether to place it on the normal queue, or the priority “VIP”. The administrator’s computer unit has full control of the queue and devices/units connected in the system - having the ability to view, edit, and delete the queues in the database, and control the flow of operations and processes. The software will be in the form of a web application, and shall therefore be compatible to almost all devices with web browser and internet connectivity, including mobile devices. Queue controller can only call the next number in queue, as well as retrieve the information of the client to be served. Moreover, the queue controller has the capability to send SMS notifications regarding the status of the clients and delivers them to their mobile phones. The system shall only support local carriers, thus, carriers of other countries shall not be accommodated. Also, the delays that might happen when sending an SMS due to network issues shall not be covered by the system.

Clients with VIP access are prioritized and will be placed automatically on the higher priority queue. For the presentation of this system, concerns regarding latecomers will not be covered; however, policies regarding latecomers shall be administered by the implementing organization.

Overall, the system is only capable of getting and retrieving requests from users, processing the request, placing it on respective queues - whether it be a normal queue or a VIP queue, print tickets as requested, continue the flow by calling the next in line, send SMS notifications to its client as needed, display the number on the screen when called, delete the request from the queue database after the transaction is complete, and shall do nothing else but the activities stated preceding this sentence.

# 

# Chapter 2 Review of Related Literature, Studies, and Systems

## Related Literatures and Studies

“Waiting is frustrating, demoralizing, agonizing, aggravating, annoying, time consuming and incredibly inexpensive,” the Federal Express once stated.

From the article *Why Waiting Is Torture* (2012) by Alex Stone, roughly 37 billion hours each year of the Americans were spent waiting in line which causes stress, boredom and that nagging sensation that one’s life is slipping away. However, according to Rabbi Carl M. Perkins (2012) in his review of the said article, waiting is distressing not because of the time you spend waiting but because of doing nothing.

Waiting in line is already evident in our everyday lives. It is observed in many service settings where capacity is fixed as peak-time demand can exceed the available supply (Kokkinou & Cranage, 2013). Furthermore, results showed that mean scores for parameters of psychological morbidity worsened during the period of waiting and improved after the procedure or transaction (Barclay et al, 2013). Circumstances of people waiting in line hold a general principle: the experience of waiting is defined only partly by the objective length of the wait. Occupied time feels shorter than unoccupied time. Research on queuing has shown that, on average, people overestimate how long they’ve waited in a line by about 36 percent (Stone A., 2012). Besides, disregarding solutions on managing queues can lead to the combination of lost revenue and ill-will (Batt & Terwiesch, 2013). Likewise, David Maister in his theory *The Psychology of Waiting Lines* (1985) mentioned that waiting-line experience in a service facility significantly affect our overall perceptions of the quality of service provided. The psychology of waiting lines examines how waits are experienced and offers managerial advice to service organizations for the tools and techniques which they could use to influence the customer’s waiting time experience.

As a result of the never ending issues regarding waiting lines, technology offers solutions to it. From the article *Long Wait Times Harming Retail Experience: Survey Says* (2012)by Rieva Lesonsky, 87 percent of consumers are willing to use some type of technology to keep them from waiting in line and 67 percent would prefer to use online check-in or down an application that saved their places in line at a retail business.

*Queuing Theory*

Queuing models have found widespread use in the analysis of service facilities, production and many other situations where congestion or competition for source resources may occur (Chowdhury, 2013). Behind these queuing models is a theory called the *Queuing Theory* discovered in 1900s by a Danish mathematician named Agner Krarup Erlang.

Queuing theory is the study of waiting lines (queues). This theory is applicable to a system with constant arrival of units (customers) and a specified number of servers (service facilities). It is used in measuring the flow of demands into and out of the queuing system, hence is also a tool to make decisions on the minimum number of resources needed (Thomopoulos, 2012).

Erlang, in his early works, stated the main elements of queuing theory: the (probabilistic) arrival process of requests (calls), the (probabilistic) service process of customers and, consequently, the departure process of customers, rejected/waiting customers, servers and etc. (Lakatos, Szeidl & Telek, 2013).

According to Lakatos, Szeidl and Telek (2013), the mathematical description of queuing systems must be in accordance with the following elements:

* Arrival process: the current system’s properties, e.g., the number of customers must be considered as customer arrivals might depend on it. But for some basic queuing models, the arrival process is identified by the interarrival time distribution.
* Service process: customer service is relevant for it could also depend on the current system’s properties and for the basic queuing models the service times are i.i.d. random variables.
* System structure: commonly the number of servers and the size of the waiting room. Thus, it talks about the resources of the queuing system.
* Service discipline: the service order and service mode of customers. It could be FCFS (first come, first served), FIFO (first in, first out), and LIFO (last in, first out). Since there are different types of customers that could arrive at the system, service order is important. In this case, to provide faster service to one customer type, priority, with or without preemptions, will be helpful.
* Performance parameters: computing for the necessary performance parameters is essential in building a precise model of a system. System utilization, mean and distribution waiting time, loss probability (the probability that a customer will be rejected by the system), etc. are the most common performance parameters.

*Computing the Waiting Time*

Queue length is one of the most crucial performance measures for signalized intersections, which is also critical to signal optimization. It has been a long-standing research topic to estimate the average queue length using loop detector data and signal timing information. Many early studies assumed discrete arrivals and integer cycle lengths, and Markov chain or similar statistical analysis techniques were applied to estimate the mean or distribution of queue lengths proposed a scheme to estimate the average queue length of a fixed-time signal by assuming traffic flow and signal timing parameters are continuous variables. Since then, queue length estimation methods can be generally grouped into two categories (Liu et al., 2009): input-output models and shockwave models. The former derives queue lengths from cumulative arrivals and departures of an intersection, while the latter looks at how the queue forms and dissipates at the intersection.

InfraRed Integrated Systems Ltd. (Irisys) specialises in the development and manufacture of intelligent infrared detectors used in, people counting, queue management, thermal imaging, healthcare and security applications (Morrison, 2010).

Utilising unobtrusive ceiling mounted infrared detectors over checkout lanes, together with detectors counting the number of people entering the store, the Irisys system provides real time information on numbers and queuing behaviours of customers. Compared to other options, such as video or beam based people counting, the unique infrared technology detects customers by their body heat, thus protecting their privacy, and providing more accurate data. Available on either PC or PDA mobile device, the Irisys system dashboard displays the average queue length, average wait time, and overall store checkout performance in real-time. The Irisys predictor software will then calculate and display how many checkouts will be needed in 15 and 30 minutes time to meet customer demand, providing in-store management with advanced notice on when they need to open and close checkouts, ensuring performance queue time metrics are met, and personnel are deployed in the right place at the right time (Clayson, 2010).

Moreover, indicators such as average waiting time for customers for a certain service since morning and in the light of average waiting time of the previous days are factors that could help in estimating the waiting time for the service selected by customers (www.rsiconcepts.com).

*Queue Management System*

Queue Management System’s top objective is to provide a better quality of service to customers. In its most basic and common form, a queue management system will issue a queue ticket to an arriving customer and later call the ticket when service is available. Customers can now sit comfortably or engaged in constructive activity while waiting for their ticket to be called thus queue management systems help to provide comfort as well as fairness to customers ([www.gms.com.m](http://www.gms.com.my)y).

Furthermore, aside from its benefits to customers, queue management system offers different kinds of advantages to the agents and to the management. The agents will have a clear picture of the customers in the queue waiting to be served through the figures shown in the system and these figures can be updated on real time clearing updating the agents about the change in flowing in and out of the organization. Moreover, the system is capable of keeping history of previous works sessions of each agent and the agent can go through this history to access the number of tickets they have served and their average serving time. Having an access to the number of tickets they have served and to the average serving time will allow agents to assess their performance and analyze their efficiency while serving different services. Managements allocate budgets for the queue management system for them to have powerful indicators in the form of reporting which can be used to improve the effectiveness and efficiency of the business operations internally and externally in terms of dealing with customers, suppliers and other business associates ([www.rsiconcepts.com](http://www.rsiconcepts.com)).

*Queue Management System Applications*

Today, queuing theory and simulation provides basis for quantitative analysis and improvement of the customer flow management processes (Grabis, 2013). It is now considered as an important branch of operations research and can be observed in different fields (Thomopoulos, 2012).

Queuing management system is also into healthcare delivery systems, where patients are the customers and either outpatient clinics or diagnostic imaging centers or hospitals are the service facilities (Gupta, 2013).

In the Philippines, BPI Express Assist, or BEA of the Bank of the Philippine Islands is one example of a queuing management system. From the article *BPI rolls out new service* by Vanessa B. Hidalgo (2011), BEA is a system that allows bank customers to enter transaction data and needed service (cash and check deposits, withdrawals, bills payment, BIR and SSS payment) on touch-screen machines. Also, the system generates a queue number and customers can sit comfortably while waiting for their turn.

*Short Message Service (SMS)*

Short Message Service (SMS) is a text based message on mobile network, pioneered by Friedhelm Hillebrand (Mitatha et al., 2011). It is used to communicate information via mobile devices. It has a limit of 160 characters which was derived from analysis of postcards, random sentences, telegraph messages, and concerns bandwidth; and these characters are transmitted from the sender to the recipients via SMS centers. Due to its exponentially growth, SMS is used in different fields including retail, banking, calendar reminders and others (Joseph, Lee, 2013).

Since SMS are fast and they deliver time sensitive information, it has been utilized as a notification system. The International Business Machines Corporation (2012) defined SMS notification as the ability of a mobile device to receive notifications as SMS messages that are pushed from a server.

*Mobile Phone Users in the Philippines*

Growth in the mobile segment of the Philippines telecom market has been moderating after years of strong expansion. Initially the slowing in the market was partly due to a general downturn in the Philippine economy; however, even as the economy picked up, it became obvious that mobile subscriber numbers were continuing to increase but at a generally slower pace. Mobile revenue growth also down in what has for many years been a highly competitive market. Into 2013 the Philippines had over 100 million mobile subscribers (Evans, 2013).

It has become almost a standard for Filipinos to utilize the SMS capability of their mobile phones. Compared to most countries, the Philippines has high access and usage statistics: an average of 101 cellphone subscriptions (both postpaid and prepaid) per 100 people; 96% of total subscriptions are prepaid; and 99% of the country is covered by a mobile-cellular network. Also, results showed 97% of Filipino mobile users use SMS. (Suarez 2012).

A new survey from consumer research firm Nielsen has found that the Philippines has the lowest smartphone adoption rate from among the Asian countries it polled recently. The study revealed only 15 percent of mobile users in the Philippines are using smartphones while the rest are using feature phones. (Nielsen, 2014)

## Related Technologies

*Context-Bounded Analysis of Concurrent Queuing Systems*

Multithreading is an important characteristic of server applications, especially for Java servers due to the lack of effective mechanisms in the current Java to share threads among client connections. In this paper of the proponents, they present the detailed workload characterization of two multi-threaded Java server benchmarks. With increasing number of threads, the instruction behavior improves due to increased locality of access. All the aforementioned miss rates and stalls drop. Resource stalls in the processor increase with increased number of threads, and eventually exceed the diminishing I-stream stalls. When there are more connections/clients, the number of instructions that the processor executes per unit of work (e.g. one transaction) increases, which affects the scalability of the system negatively. Further enhancement in the operating system and the Java virtual machine may be expected to alleviate this problem (La Torre, Madhusudan & Parlato, 2008).

*Red Oxygen: Crunchbase Review*

Based in San Francisco, Red Oxygen is one of the world's leading developers and distributors of enterprise SMS software applications and services. Their tools combine the power of email applications, instant messaging clients, and CRM applications with the convenience and mobility of text messaging; enabling you to reach your clients and colleagues near-instantly with time-sensitive updates and reminders.

Red Oxygen's products allow users to send SMS to specific individuals or broadcast SMS to entire lists. Send SMS to recipients' mobile devices from your email client and receive replies back in your Inbox. Or schedule SMS to automatically send out notices/calendar reminders at a specified date and time. The software is available for Microsoft Outlook, IBM Notes (formerly Lotus Notes), IBM Sametime, and Windows, as well as web-based and CRM custom integration.

Red Oxygen has clients across a broad spectrum of industries. Our current client list includes such corporations as Microsoft, Panasonic, General Electric, Pfizer, and AT&T, among countless others. They've chosen to use Red Oxygen products and services to reduce telecommunications expenses, improve the speed at which their messages are read, and leverage existing communications infrastructure.

With a decade-long track record of success — 300,000 users in over 50 countries, numerous Fortune 10 clients, & more than 100 million messages sent & counting — we’re trusted experts in the competitive sector of enterprise messaging. (Sheahan et al, 2013)

*IPFrontline Development with Project Red Oxygen*

With traditional BPM solutions typically taking up to 18 months to roll out, a company may find that its needs have changed and the solutions have become outdated by the time they are implemented. This challenge is exacerbated by ever-expanding volumes of data and new ways to reach internal and external stakeholders. OpenText's Process Suite optimizes operational processes and provides more flexibility, through packaged applications, pre-built delivery solution components, and tools that cut the time spent designing and deploying solutions.

Automating business processes across an organization has proven to be a challenging task for companies, which often experience a mismatch between the cycle of business needs and the available supporting technology, the market needs a solution to better manage workflow to help increase their productivity and cut development times. (Sinur, 2014), noted process expert and former Gartner analyst.

Process Suite is part of OpenText's Project Red Oxygen, which enables organizations to discover and manage information to spur growth and innovation across their organization, and increase time to competitive advantage.

Project Red Oxygen increases time to competitive advantage by providing five comprehensive and integrated product suites, including Process Suite, Experience Suite, Discovery Suite, Information Exchange Suite, and AppWorks. Together, they deliver a unified developer platform, an engaging and consistent user experience and deep integration across suites to enable CIOs to partner with all lines of business.

# 

# Chapter 3 Technical Background

## 3.1 Design Methodology

The team decided to use Agile Development Model as an appropriate methodology for creating the system, as it allows the team to develop the system through small increments. The method is flexible, and can easily adapt to new requirements.

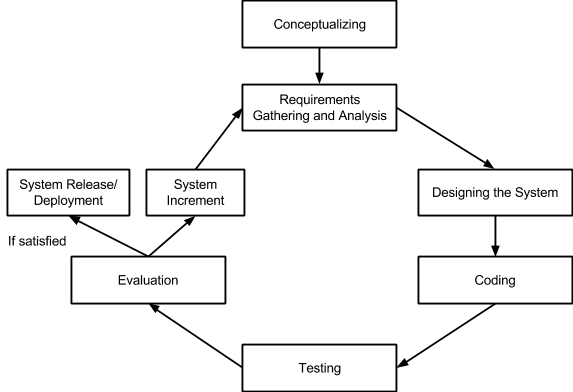


Figure 3.1.1: Agile Development Model

In the conceptualizing, the proponents will research on the present queuing management systems and will base the abstraction of the proposed system to it. The proponents will identify the deficit seen in the systems researched and use the findings to aid in improving the proposed system.

As for the requirements gathering and analysis, the proponents took into consideration the investigation of the essentials needed to complete the system such as the touch screen LCD, a dedicated thermal printer, a controller, and a server. Furthermore, the proponents will research on different SMS modules available to handle SMS operations in the most efficient and easiest way.

## 

## 3.2 Requirements Analysis

### 3.2.1 Functional Requirements:

* The kiosk should allow clients to choose their desired transaction type
* The web application shall ask clients to enter their mobile number and an option whether the client should receive SMS notifications
* Clients must be enqueued on the normal queue whenever they receive a number
* VIP clients must be enqueued on the higher priority queue whenever they receive a number
* System will print a number for client as requested
* The server must send SMS notifications to the clients as indicated
* Controller can call the next number, as well as 1-step previous number for undo function
* Controller must flash the called number on a separate screen visible to the clients as well as update the real-time module of the web application
* Speaker should play a notification tone
* Database must dequeue the number after transaction

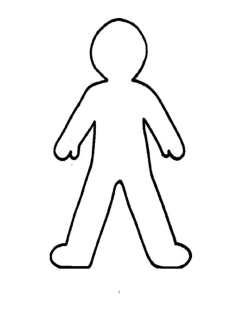
### 3.2.2 Non-Functional Requirements:

* Controlling the queue should be instantaneous
* System must support more than one controller
* Maintainability of the transaction options doesn’t require hardcoding or source code modification
* System has the ability to recover from power failure by keeping a nonvolatile backup copy of the queue

## 

## 3.3 Overview

The structure of the system shall consist of four units: the mobile devices of the users or the kiosk machine for clients’ requests, a controller for calling the next number, a server hardware which runs the web server application and the database application which processes all data, and a display screen which visually shows the sequence of activities in a process and the flow of the system. The flow of the system will start upon entering the client’s mobile number on the specified field - enabling the client to automatically receive an SMS notification which confirms his request. The system would also ask if the client would like to receive updates of the queue status via SMS notifications; if not, the client should then monitor the status of the queue on the real time module of the web application. Also, if the client happened to be a privileged VIP guest, client’s username and password must be entered. Printing the ticket will be available only to clients without smartphones – they will be using the kiosk to process their requests and have an option to receive SMS Notification. Without SMS Notification, the ticket will be printed without the need for further user input.

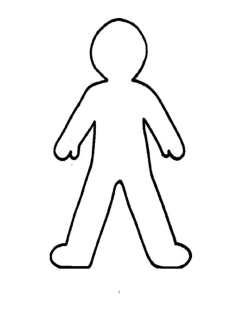


Client

An SMS notification confirming client’s request will arrive in his/her mobile phone on the time specified. He/She could also receive SMS updates regarding queue status

Client inputs phone number and if he/she wants to be notified via SMS. Prioritized clients input their username and password

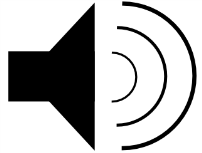
Client without a smartphone is accommodated using the kiosk which runs the web application

**

**

Business Administrator

Client without a smartphone has an option to print his/her ticket number, or be notified thru SMS

**



NOW SERVING

Counters

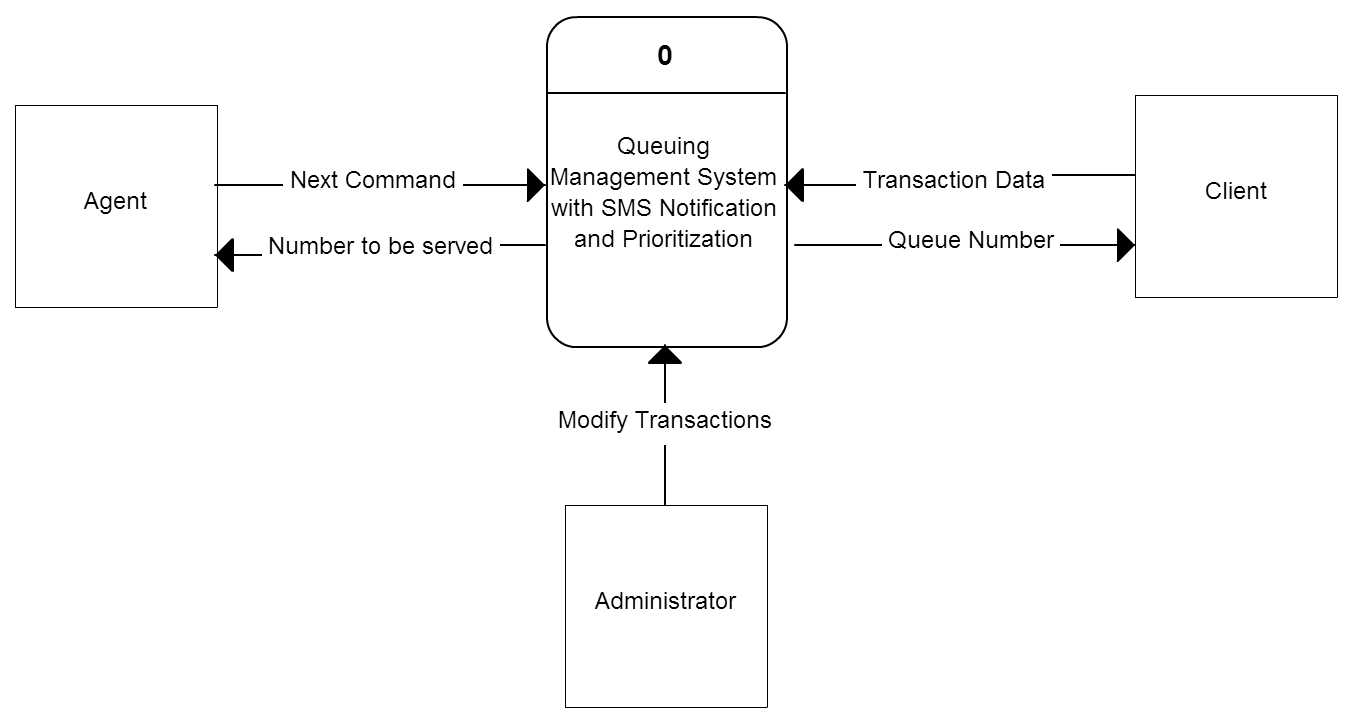
1 2 3

**N66 P07 N65**

System plays audible notification tone

Display flashes the number on screen

Figure 3.3.1: Basic system operation

Figure 3.3.2: Context Diagram (Data Flow Diagram - Level 0)

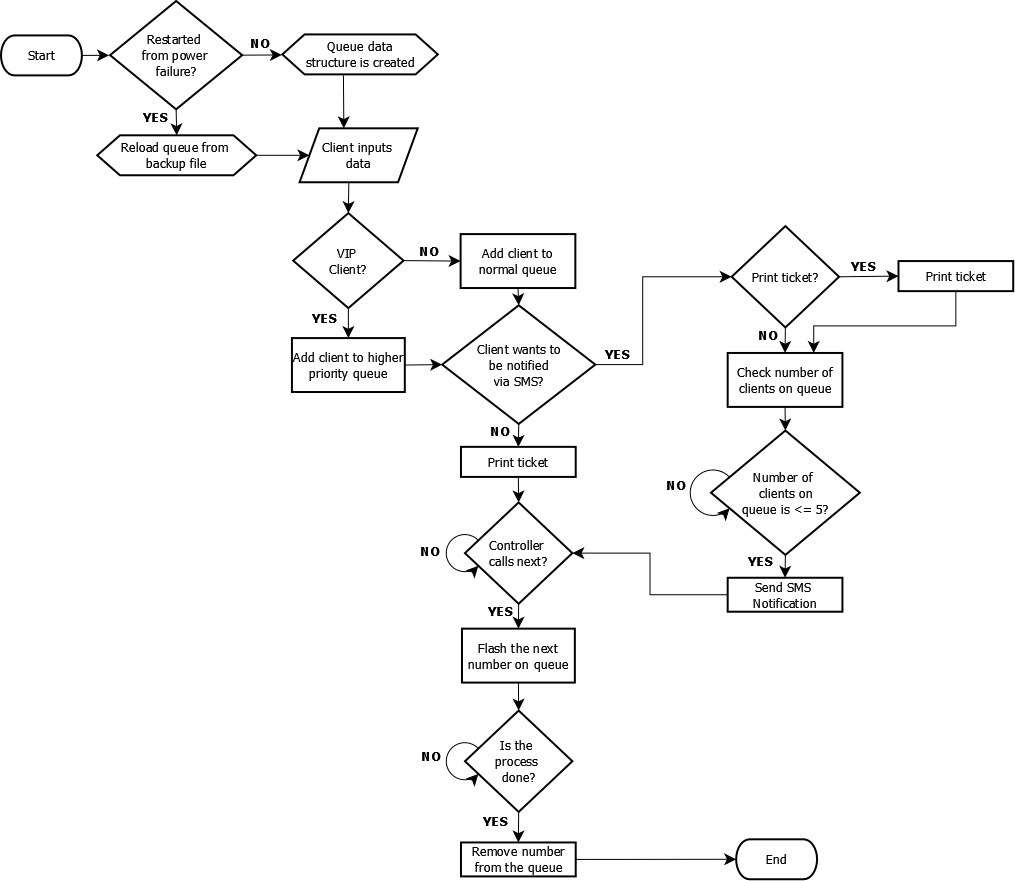


Figure 3.3.3: Flowchart of the system

### 3.3.1 System requirements

The project consists of several units, operating as one complete system. Web App, Queue Controller, and Display Desktop App will be programmed using Java Programming Language in Java EE (Enterprise Edition), JSP, Java Servlet, and Java SE (Standard Edition). Database used is Java Derby DB, an open source database server provided by Apache. Client’s mobile devices must be connected to the internet, with an internet browser to access and utilize the web application.

The database is the datacenter of the system - responsible for processing all data flowing in the system. The Java API used support server services at desktop OS level to provide server capabilities to a non-server system. The display is responsible for outputting the call request of the clients for the next person in the queue. Controllers are in desktop app interface, allowing the control of queue. A network interconnecting device may be in the form of a switch or a hub if using wired connection, or a wireless access point, if using wireless connection for this system. It creates the Local Area Network (LAN) dedicated to the system, and is responsible for bridging connections among the units within the system as a pathway of data interchange. For the Wide Area Network (WAN), a router is necessary for the server to connect to the internet, and utilize Red Oxygen SMS Gateway.

#### 3.3.1.1 Hardware Specifications

The kiosk will compose of a touchscreen LCD which serves as an input device, and a dedicated thermal printer that generates the ticket number. Moreover, the server must have at least the following hardware requirements: a processor with at least 2 cores for an effective multithreading, 2GB of RAM, 512MB disk space, and a network interface card (NIC). A LAN Card is required if using wired connection, requiring a switch to bridge the network interconnection. Otherwise, Wi-Fi module shall be installed to use wireless connection, and a wireless access point (AP) will be required to accommodate the wireless needs. A video card with multi-monitor support is also required for separating the desktop of the server, and the display interface.

#### 3.3.1.2 Software Specifications

Since the programming language to be used is Java, the software is available to run on any hardware that is capable of running Java Virtual Machine (JVM). According to Sun Microsystems, Java is *Write Once, Run Anywhere* (WORA), which means that it can be developed on any device, compiled into a standard [bytecode](http://en.wikipedia.org/wiki/Bytecode) and be expected to run on any device equipped with a [Java virtual machine](http://en.wikipedia.org/wiki/Java_virtual_machine) (JVM). JVM can run virtually on any desktop operating system where it supports. Java Swing GUI API will be used for the system’s user interface which would enable users to enter data, select buttons and interact with the system. Data flow shall start from the web application that will send all information to the database. The database will process the data, and send necessary information back to the web application, such as the client mobile number and its queue number. The web app and the controller, with the use of database, then passes the message to Red Oxygen SMS Gateway, the one responsible for sending it to the mobile phones through SMS. Recommended Web browsers includes: Internet Explorer 11, Google Chrome 38, and Safari 7.0, while other web browser vendor or older versions of the listed browsers may still support be supported, but aren’t guaranteed to perform with optimal output.

### 3.3.2 Prototype

The project prototype will initially run on the team’s personal device. Figure 3.4 shows how hardware components shall be interconnected in the system.

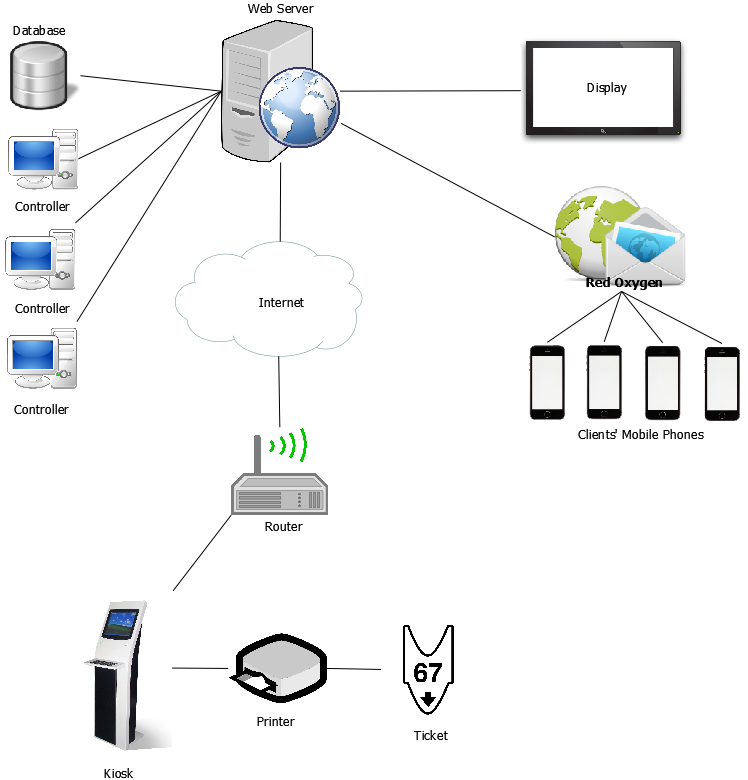


Figure 3.3.2.1: System Architecture

SMS operations are forwarded to Red Oxygen on the internet, and shall be the one responsible for message forwarding to mobile clients.

User interface and the database is provided by Java Swing GUI Framework. Below are the sample user interface designs to be implemented.

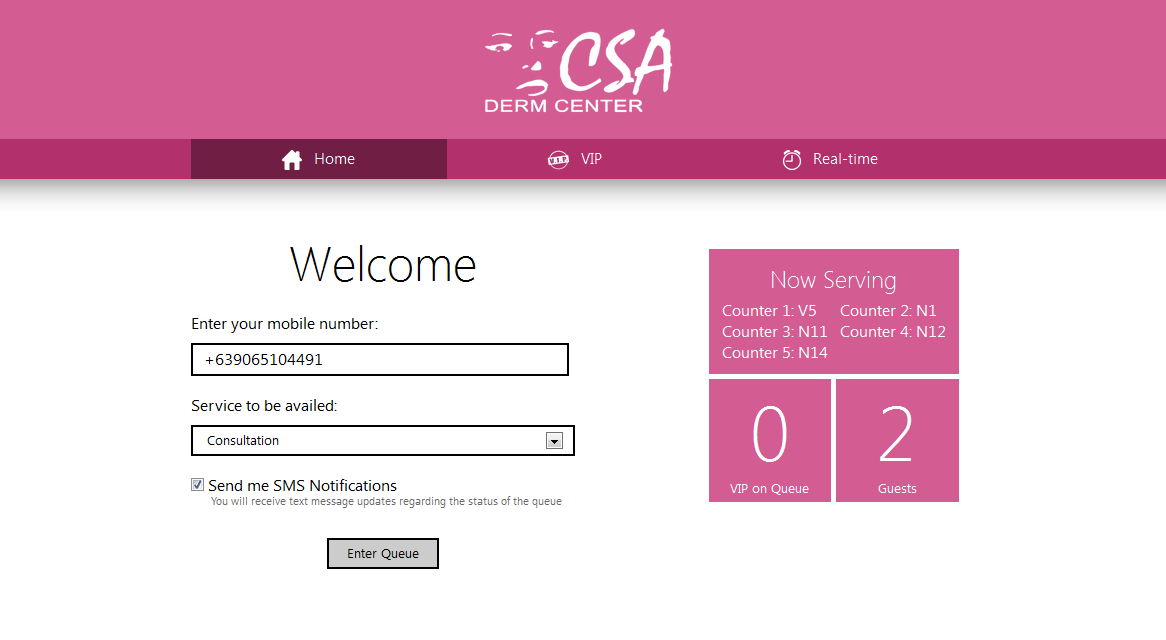


Figure 3.3.2.2: Home Screen for desktop

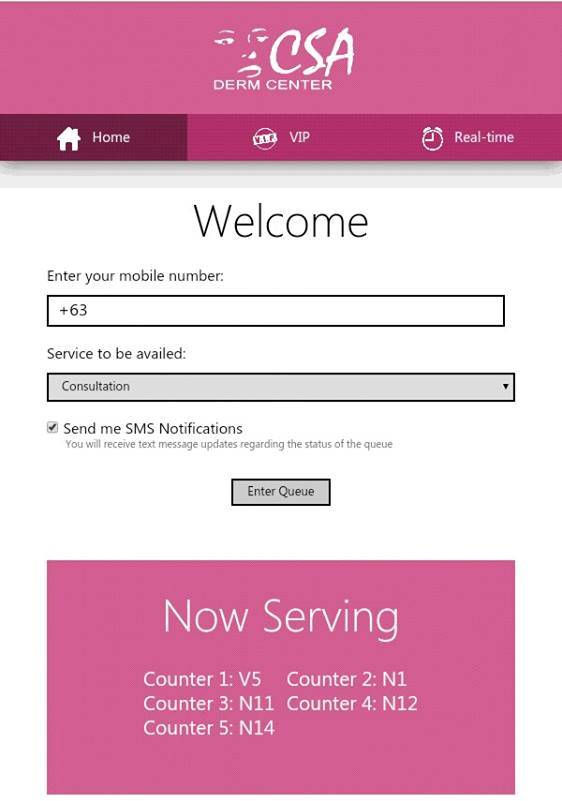


Figure 3.3.2.3: Home screen for mobile

The Home Page will be the first screen to be displayed. It shows the fields for client’s mobile number, service to be availed, and a checkbox if he/she wants to be notified via SMS. Also, a quick glance-able queue status slides up at the right side of the screen (desktop view) or bottom part of the page (mobile view) shows information regarding the currently being served number, as well as the number of regular and VIP clients on queue. Regular clients are expected to navigate this part of the web application for them to be placed on the normal queue.

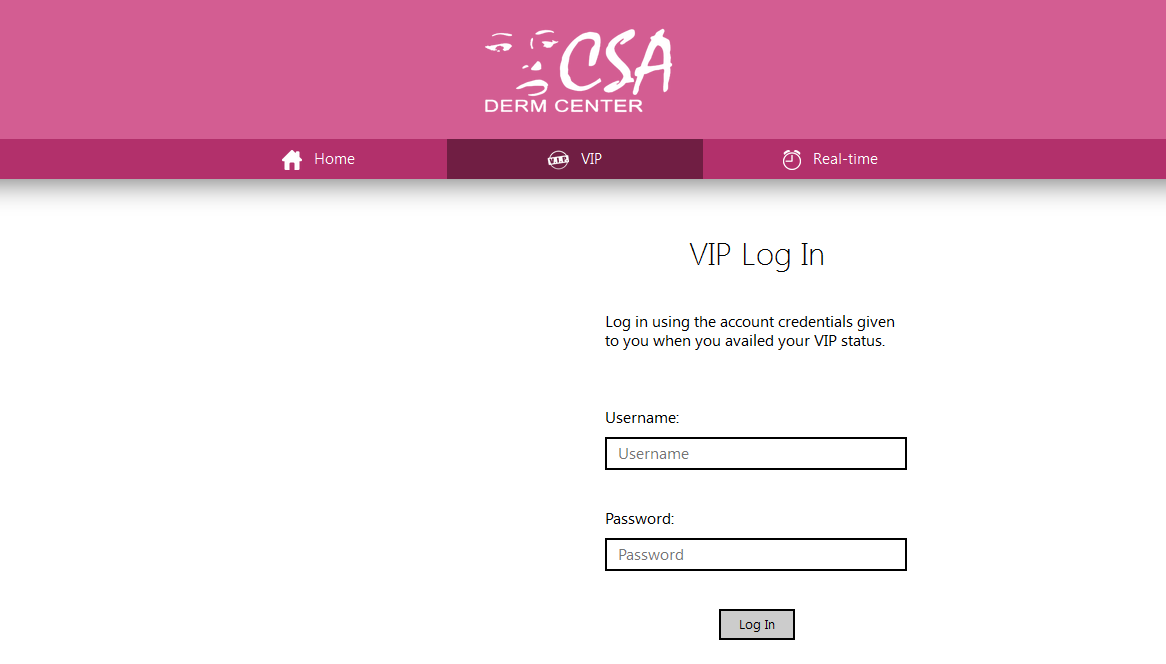


Figure 3.3.2.4: VIP screen for desktop

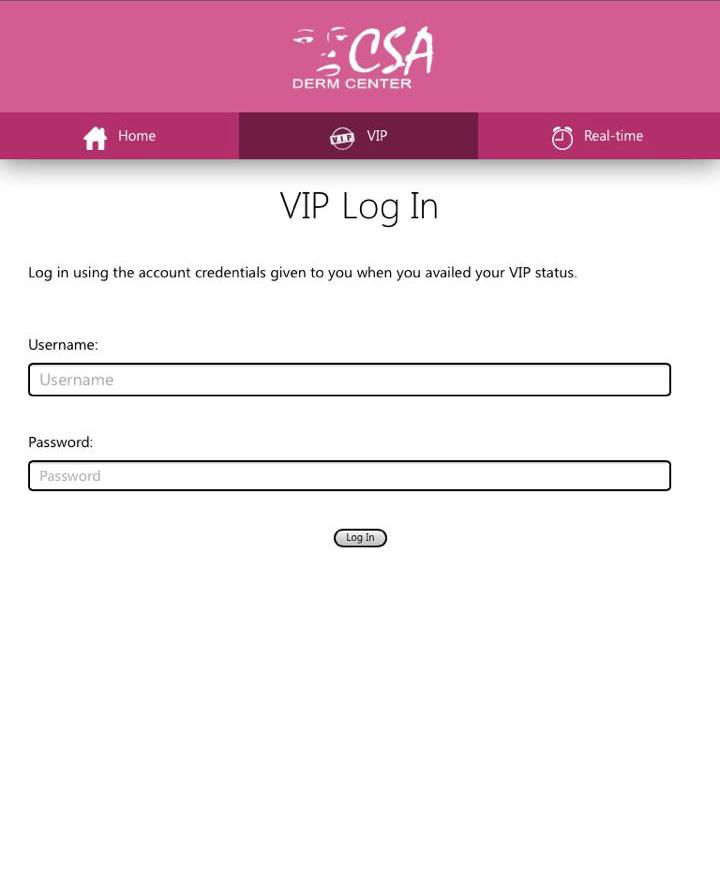


Figure 3.3.2.5: VIP screen for mobile

VIP members can access higher priority queue by accessing the VIP tab and by entering their own username and password administered by the implementing organization.

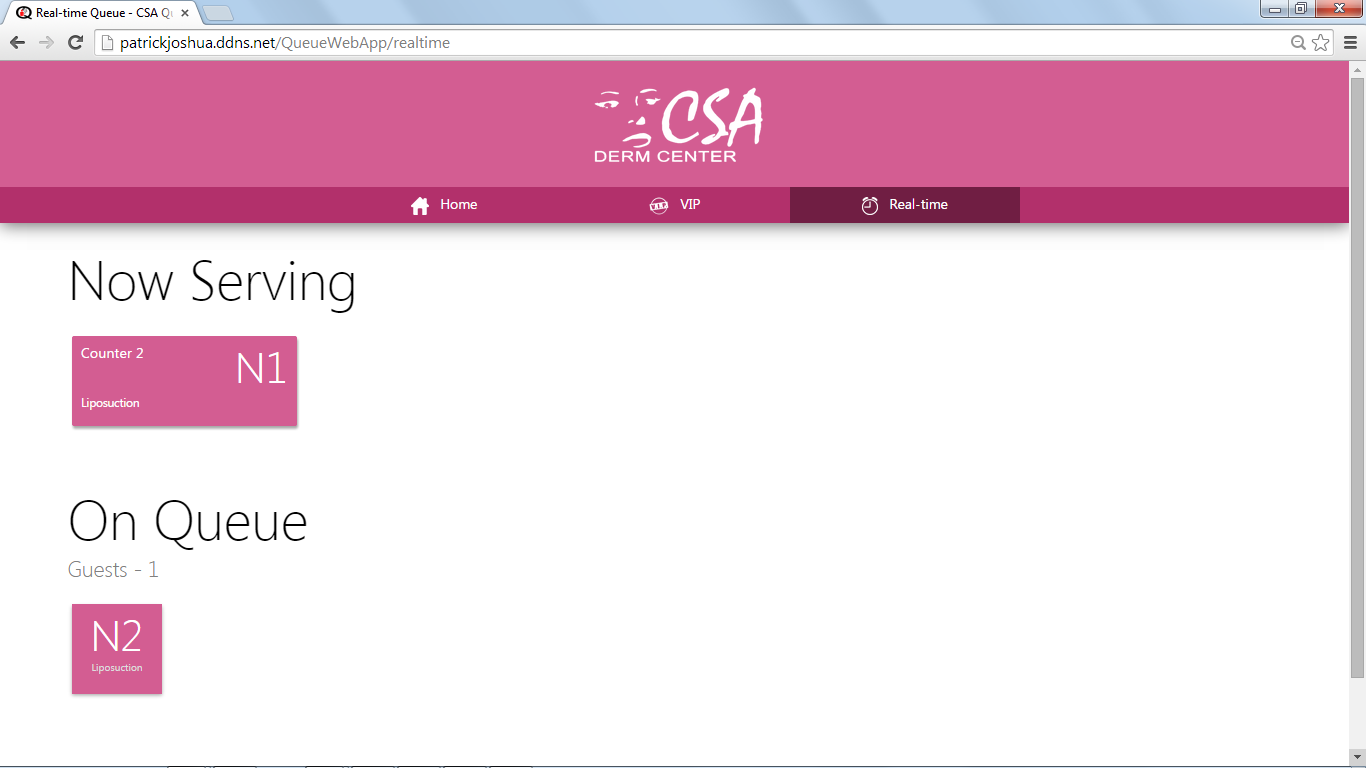
**

Figure 3.3.2.5: Real-time screen for desktop

This feature of the web application will help clients to visualize the current status of the queue. It shows queue numbers that are currently served, allowing clients to monitor their queue status.

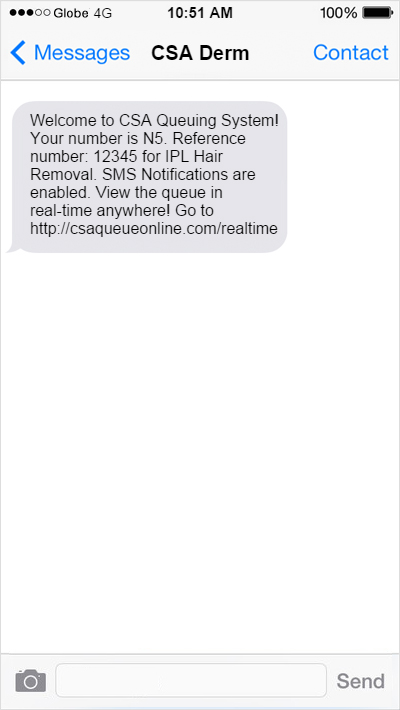
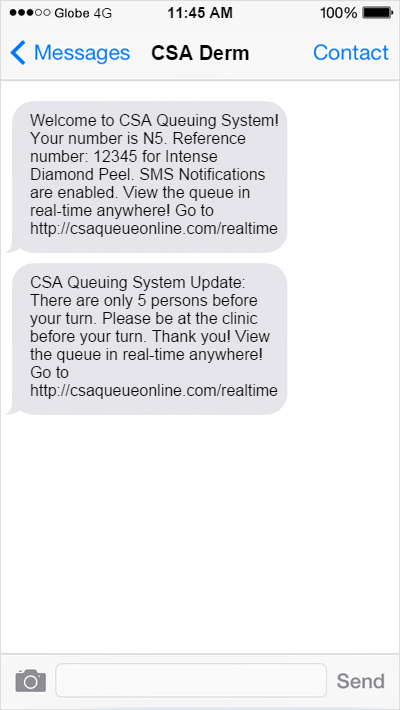


Figure 3.3.2.7: SMS Confirmation & Notification received by the Client

The image above emulates a user interface seen only by the client generated by the system containing information regarding the queue number, a reference number, and the client’s request. Another message will be sent which contains how many persons left the client's turn. The template is produced in a user friendly and straightforward format.

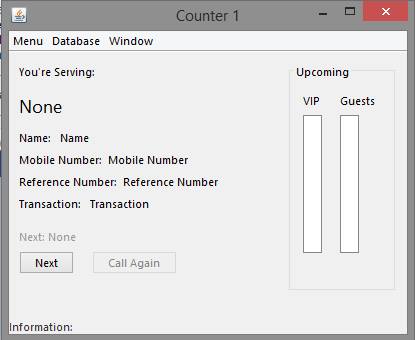
**

Figure 3.3.2.8: Controller interface

The interface above is only seen by the administrator, mainly for monitoring purposes - shows the timestamp and information entered into the database, shows the real-time service to the clients via their respective queue number as well as the queue status. This controller is used to utilize all the features of the system. Examples of system’s features are displayed in the figures below:

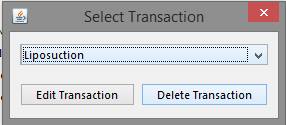
**

Figure 3.3.2.9: Updating or modifying transaction list

This screen allows the administrator to modify the transaction list’s i.e. to change transaction’s order and position, edit a transaction, or delete a transaction from the list.

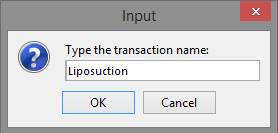


Figure 3.3.2.10: Adding a new transaction

A pop-up menu that allows administrator to input a new transaction name to be added on the list.

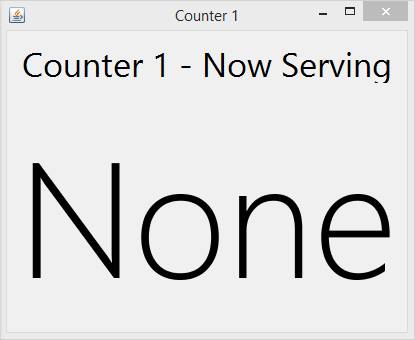
**

Figure 3.3.2.11: Display screen

## Chapter 4 Implementation, Results and Discussion

### 4.1 Requirements Documentation

#### **4.1.1 Development Methodology**

The proponents decided to use the Agile Development Model approach as an appropriate methodology for the system. It could help the team satisfy its goals and objectives by developing the system in small increments. The method is flexible that new features are delivered quickly and frequently, with a high level of predictability. Also, its iterative lifecycle and incremental nature makes any sorts of changes much less disruptive compared with other methodologies.

The first phase is the conceptualizing; the proponents studied the nature of present queuing management systems and based the abstraction of the proposed system to it. The proponents checked on the strengths and weaknesses of the systems researched and used the findings to aid in improving the proposed system.

The requirements gathering and analysis phase was done for the investigation of the essentials needed to complete the system such as the touch screen LCD, a controller, and a server. The proponents observed the behavior of different queuing management systems by visiting banks, hospitals, and other business establishments that integrate the said system in their daily operation. Moreover, requirements gathered during this stage guided the proponents to determine the features of the project and to draw the boundary line. At this point of study, the proponents analyzed the scope and limitation of the project that could be found on the first chapter of this project.

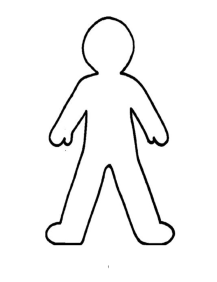
The next phase which is designing the system comes after having the right concept and when there is enough requirements to create the system. During this phase, the proponents visualize how the project will look like and what will be the core processes and modules. The proponents used different web development tool and photo editing software such as Adobe Dreamweaver and Adobe Photoshop to help the team envision the design of the system. This is a substantial phase of the development as the proponents need to picture out the project and how the design shall be executed.

After designing the system and how data will be managed, coding comes next. Back-end and front-end developers collaborated to translate the conceptualized and designed system into a programming language in a specified platform. Additionally, the proponents made the best possible manner on how to implement the design of the system such as reducing the testing and maintenance effort. To achieve these, the proponents used Java as the programming language and Apache Derby as the database. Since the project is web based, the proponents used Hypertext Markup Language (HTML), JSP for the development of the website. Additionally, Adobe Photoshop was used for the web banners and other designs of the system.

Testing is the next phase after a successful coding. This phase verified the functionalities of the system. The system was tested to see if a feature is not functioning properly. Reviews, inspections, and walk-throughs were done during this testing to check whether the system meets all requirements. Additionally, this phase includes working with the software, giving input values and checking if the output is as expected.

The proponents evaluated the system after exposing it to different tests wherein errors were found and fixed. The proponents observed the system in use, looked at the solution’s recent performance over time. This phase defines the areas for improvements which is essential in

#### **4.1.2 Event/Use Case Diagrams**



<<includes>>

<<includes>>

<<includes>>

<<includes>>

Figure 4.1.2: Client use case diagram

Figure 4.1.2 shows the use case diagram of the Client. The use case describes the capabilities of the client in utilizing the system.

#### **4.1.3 System Requirements**

Minimum Configurations:

* For Microsoft Windows 7 or 8:
  + RAM: 128 MB
  + Disk space: 124 MB for JRE; 2 MB for Java Update
  + Processor: Minimum Pentium 2 266 MHz processor
* Software and Peripheral:
  + Browsers: Internet Explorer 9 and above, Firefox, Chrome, Safari
  + Internet connection
  + Java Runtime Environment 6

Recommended Configurations:

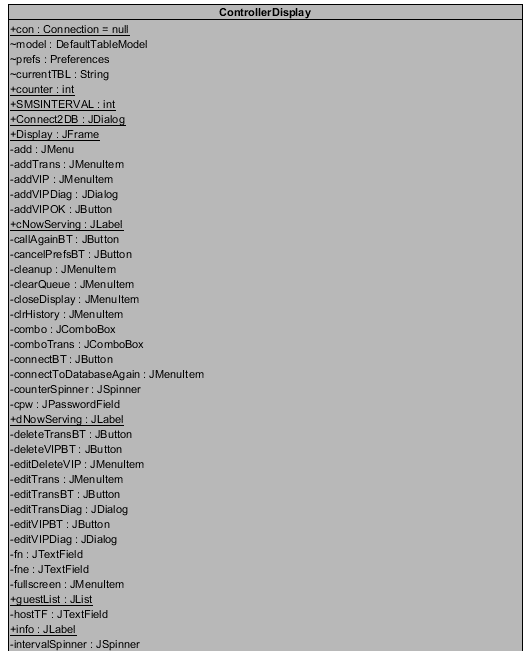
* For Microsoft Windows 7 or 8:
  + RAM: 128 MB
  + Disk space: 124 MB for JRE; 2 MB for Java Update
  + Processor: Minimum Pentium 2 266 MHz processor
* Software and Peripheral:
  + Browsers: Internet Explorer 11, Chrome 34, Safari 8.0
  + Fast Broadband Internet connection
  + Java Runtime Environment 8

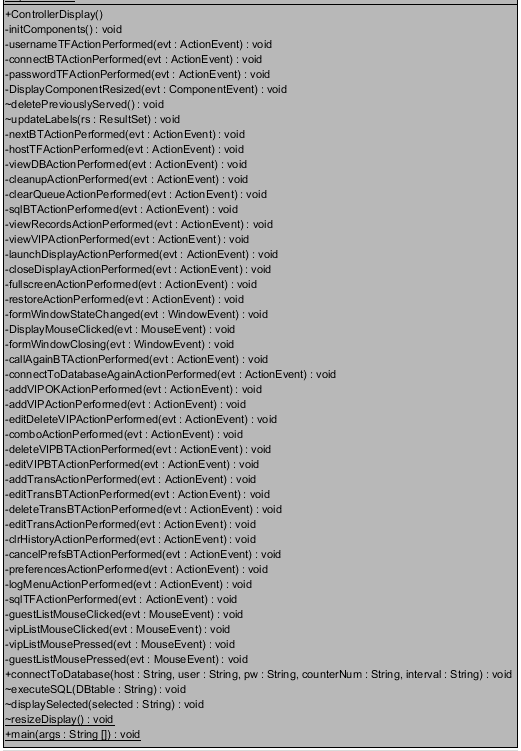
Hardware and Software

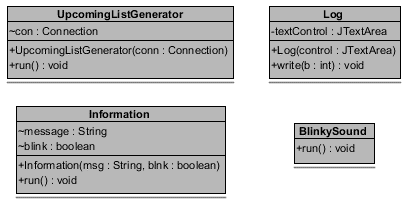
To use the system, at least one desktop or laptop computer is required, with a desktop or server operating system, and basic peripherals. Windows, Mac OS X, or Linux may be used, either in 32 or 64 bit architecture. Internet connection is required for SMS interface, and web app hosting. The hardware must also support multiple displays of at least two output to separate controller’s main interface from the display. Any server application will do, but recommended software is Glassfish server from Oracle. Same for the database, any database server will function, but the tested database server is Apache Derby DB.

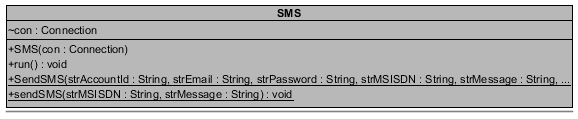
To run the database server on Windows, a TCP/IP Protocol Support and JDBC Driver for the JDBC Connection are necessary.

**4.2 Design of Software, Systems, Product, and/or Processes**

**4.2.1 Class Diagrams**





Figure 4.2.1.1: Unified Modelling Language Class Diagram

#### **4.2.2 Sequence Diagrams**

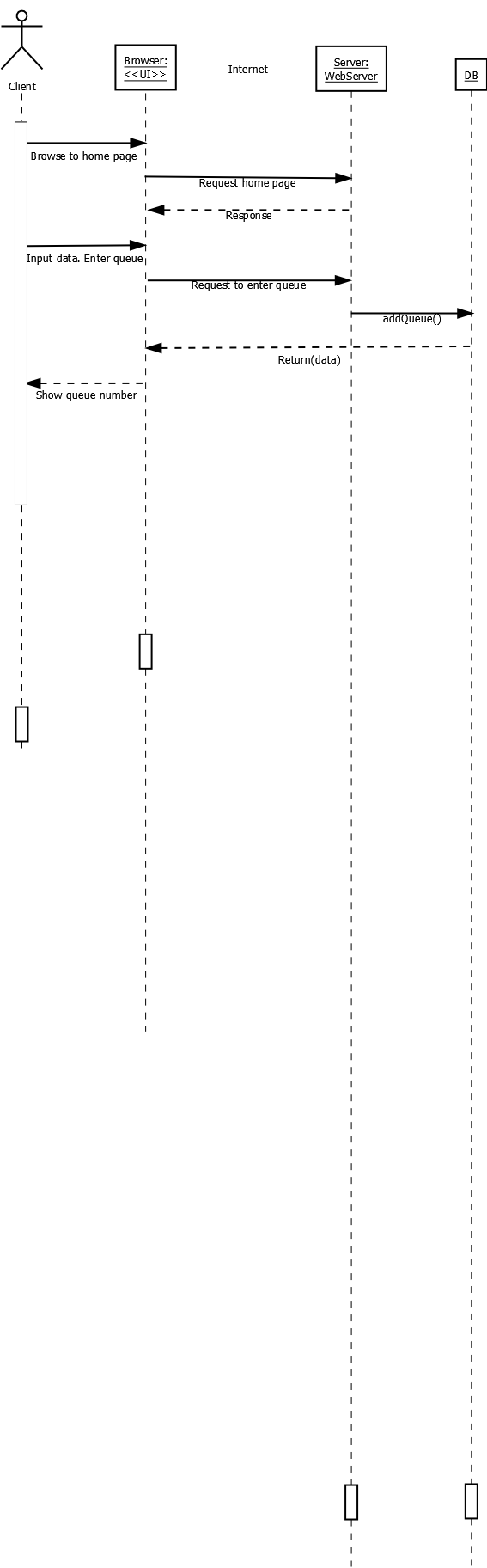


Figure 4.2.2.1: Home page sequence diagram

Figure 4.2.2.1 shows the sequence of events that will take happen when a client connects to the web application to process his request. First, for a client to view the home page, he must use a web browser that would display the user interface. The web browser will send request to the server to load the homepage using an internet connection. The server will respond by displaying the home page where a normal client enters necessary information to enter the queue. Once a client has provided all the necessary information he will be placed on the normal queue which resides in the database. A message will be sent to him confirming his request. The database will generate a text message sent to the client’s mobile phone when the remaining number of persons on the queue is at least five.

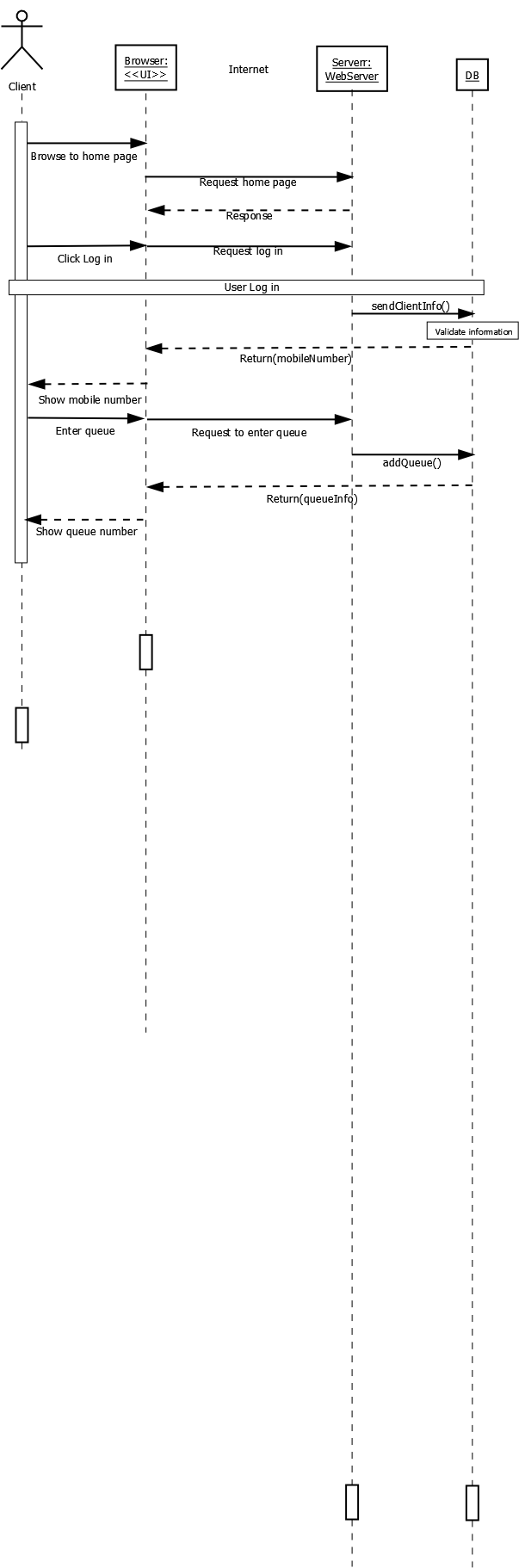


Figure 4.2.2.2: VIP page sequence diagram

A VIP Client will log in using his username and password which can be found on the VIP tab of the web application. After a successful log in, the system will ask for the VIP Client’s mobile number in the database and automatically writes it in the mobile number field. After choosing a request and selecting or deselecting the checkbox for SMS notification, the system will place the VIP client on the higher priority queue. The system will generate a text message sent to the VIP client when the remaining number of clients on the queue is at least five.

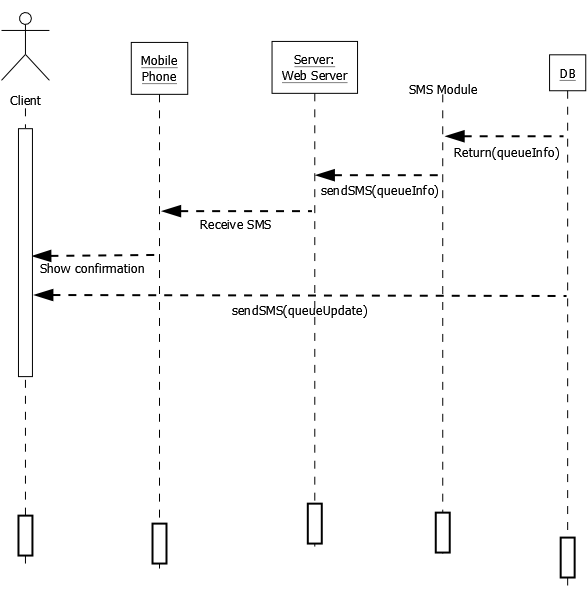
**

Figure 4.2.2.3: Receiving SMS notification sequence diagram

Clients regardless of their priority receive an SMS confirming their request in entering the queue. Also, an SMS notification about the queue status will be sent to clients who availed for it. This can be observed when clients’ request is successfully entered on the queue. The database will pass the contents of the message to an SMS module. The connection between the SMS module and the web server will allow clients to receive the confirmation of their request or updates regarding the queue status.

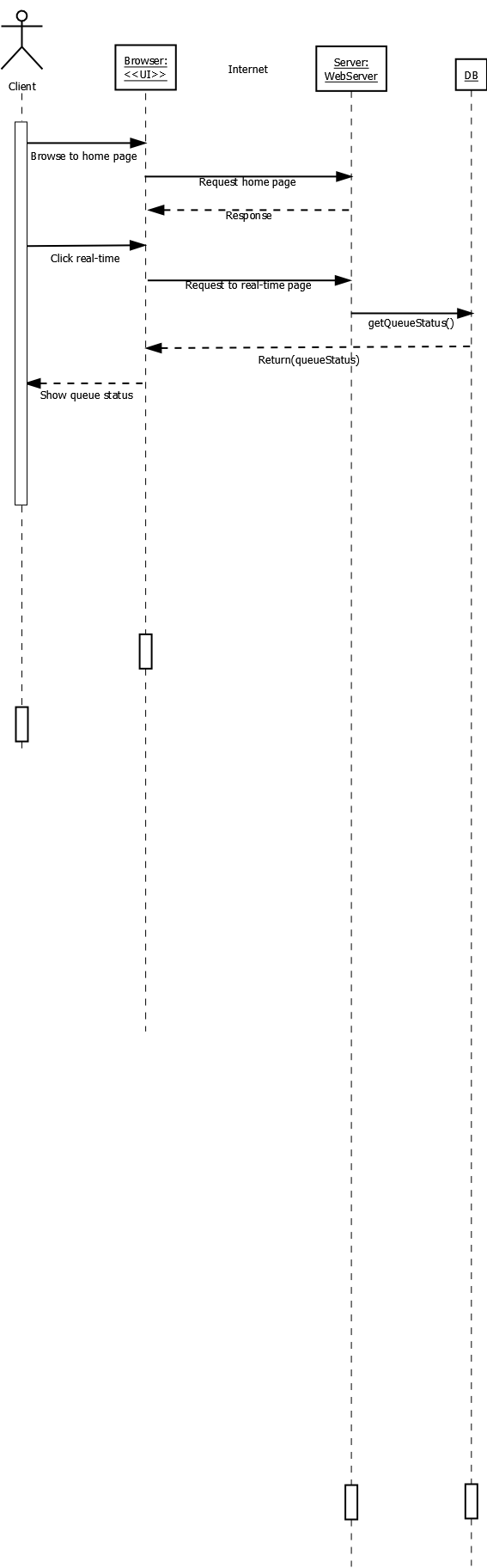
**

Figure 4.2.2.4: Real-time page sequence diagram

Clients can view a real-time status of the queue by using a web browser to show the web application. The browser will send request to the web server through an internet connection to load the web application. Displaying the real-time status of the queue happens whenever clients click on the real-time tab. Upon clicking, the browser will connect to the web server. The web server will get the queue status from the database; and pass it to the web browser that would eventually display the real-time interface of the web application.

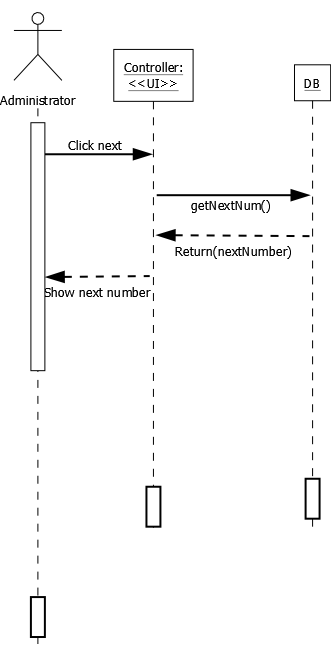


Figure 4.2.2.5: Controller sequence diagram

Figure 4.2.2.5 shows the interaction of objects and user of the controller in displaying the next number. The business administrator (the person who will use the controller) will select the next button in the controller’s interface. The controller is connected to the database and will request to get the next number from it. The database will respond by returning the next number to the controller so that the user can view it.

#### **4.2.3 Network Design**

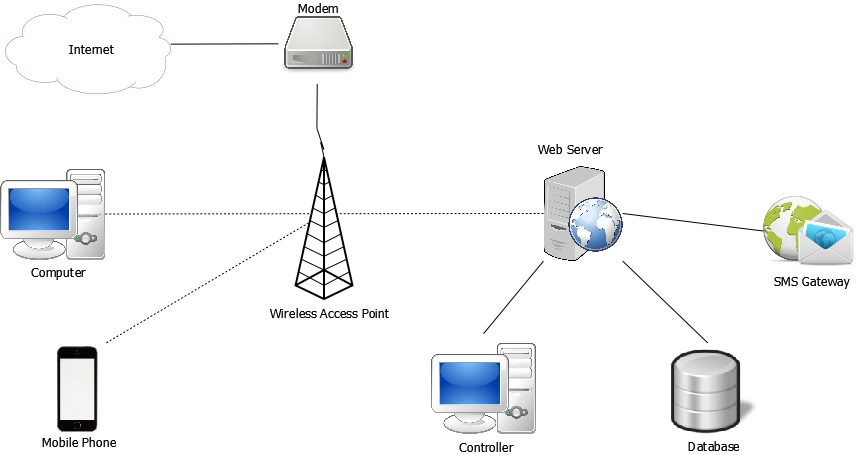
****

Figure 4.2.3.1: Network Interconnection

As shown in Figure 4.2.3.1,the network design of the system consists of eight modules: a web server where the controller, the database, and the SMS gateway connects, an internet where the modem connects, a wireless access point where a computer and a mobile phone connects.

The network design of the system shows how a client enters the queue using a mobile phone or a computer that must be connected to the internet (local or wireless) for the web application to be utilized. It also shows that the web server is the reason why every components of the system communicates with each other. Examples are: sending SMS to clients’ mobile phones, getting information from the database and displaying it to the web application, and most importantly, handling the queue using the controller.

.

#### **4.2.4 Database Design**

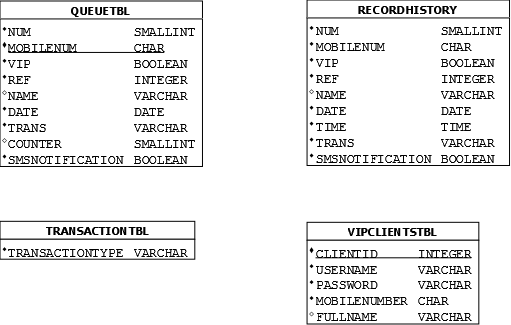
****

Figure 4.2.4.1: Database Structure

The database has a total of four tables. The QueueTbl contains all data that were entered by regular clients in the home tab of the web application. The counter part of this table is the VIPClientsTbl that stores VIP clients’ information. The RecordHistory houses every data that were entered in the web application. This could help the business in generating statistical records that could be used as baselines for improvements. Lastly, TransactionTbl contains all transactions offered by the business. Authorized personnel can modify this depending on the services offered by the business.

### 4.3 Development and Testing

#### **4.3.1 Test Case Scenarios**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.1 Home page (desktop view) – correct cellphone number with SMS notifications enabled | | | | | |
| HP1 | Low | Select textbox area and input cellphone number | Input cellphone number without delay while format pops up and disappears after input | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Check send me SMS notifications | Checked box without delay | Box checked upon opening of website | Passed |
| HP4 | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.2 Home page (desktop view) – correct cellphone number with SMS notifications disabled | | | | | |
| HP1 | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4 | High | Click Enter Queue | Receive queue number | Queue number received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.3 Home page (desktop view) – correct cellphone number with SMS notifications enabled then cancelled request | | | | | |
| HP1 | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4 | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |
| HP5 | High | Click Cancel Request | Pop up indicating request cancelled appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.4 Home page (desktop view) – correct cellphone number with SMS notifications disabled then cancelled request | | | | | |
| HP1 | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Uncheck send me SMS notifications | Unchecked box without delay | Box unchecked swiftly | Passed |
| HP4 | High | Click Enter Queue | Pop up appears showing queue status | Queue number received | Passed |
| HP5 | High | Click Cancel Request | Pop up indicating request cancelled appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.5 Home page (desktop view) – incorrect cellphone number with SMS notifications enabled | | | | | |
| HP1 | Low | Select textbox area and input incorrectly formatted cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4 | High | Click Enter Queue | Pop up indicating incorrect cellphone number format appears | Pop up appeared to re-enter correct format for cellphone number | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.6 Home page (desktop view) – incorrect cellphone number with SMS notifications disabled | | | | | |
| HP1 | Low | Select textbox area and input incorrectly formatted cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3 | Low | Uncheck send me SMS notifications | Unchecked box without delay | Box unchecked swiftly | Passed |
| HP4 | High | Click Enter Queue | Pop up indicating incorrect cellphone number format appears | Pop up appeared to re-enter correct format for cellphone number | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.7 VIP login (desktop view) – login with correct password with SMS notifications and prioritize me enabled | | | | | |
| VL1 | Med | Log in with given credentials | Log in successful, page redirected to queue entry | Log in successful and page redirected to queue entry | Passed |
| VL2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| VL3 | Low | Check send SMS notificaitons | Checked box without delay | Box checked swiftly | Passed |
| VL4 | Med | Check Prioritize me box | Checked box without delay | Box checked upon loading of site | Passed |
| VL5 | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.8 VIP login (desktop view) – login with correct password with SMS notifications disabled | | | | | |
| VL1 | Med | Log in with given credentials | Log in successful, page redirected to queue entry | Log in successful and page redirected to queue entry | Passed |
| VL2 | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| VL3 | Low | Uncheck send SMS notificaitons | Unchecked box without delay | Box unchecked swiftly | Passed |
| VL4 | High | Click Enter Queue | Receive queue number | Queue number received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.9 VIP login (desktop view) – login with incorrect password | | | | | |
| VL1 | High | Log in with incorrect credentials | Pop up of incorrect password appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.10 Real-time queue (desktop view) | | | | | |
| RTQ1 | Med | View page | Paged viewed with continuous updates | Page loaded with continuous updates | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.11 Home page (mobile view) – correct cellphone number with SMS notifications enabled | | | | | |
| HP1m | Low | Select textbox area and input cellphone number | Input cellphone number without delay while format pops up and disappears after input | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Check send me SMS notifications | Checked box without delay | Box checked upon opening of website | Passed |
| HP4m | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.12 Home page (mobile view) – correct cellphone number with SMS notifications disabled | | | | | |
| HP1m | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4m | High | Click Enter Queue | Receive queue number | Queue number received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.13 Home page (mobile view) – correct cellphone number with SMS notifications enabled then cancelled request | | | | | |
| HP1m | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4m | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |
| HP5m | High | Click Cancel Request | Pop up indicating request cancelled appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.14 Home page (mobile view) – correct cellphone number with SMS notifications disabled then cancelled request | | | | | |
| HP1m | Low | Select textbox area and input cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Uncheck send me SMS notifications | Unchecked box without delay | Box unchecked swiftly | Passed |
| HP4m | High | Click Enter Queue | Pop up appears showing queue status | Queue number received | Passed |
| HP5m | High | Click Cancel Request | Pop up indicating request cancelled appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.15 Home page (mobile view) – incorrect cellphone number with SMS notifications enabled | | | | | |
| HP1m | Low | Select textbox area and input incorrectly formatted cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Check send me SMS notifications | Checked box without delay | Box checked swiftly | Passed |
| HP4m | High | Click Enter Queue | Pop up indicating incorrect cellphone number format appears | Pop up appeared to re-enter correct format for cellphone number | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.16 Home page (mobile view) – incorrect cellphone number with SMS notifications disabled | | | | | |
| HP1m | Low | Select textbox area and input incorrectly formatted cellphone number | Input cellphone number without delay | Cellphone number successfully inputted with smooth pop up animation | Passed |
| HP2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| HP3m | Low | Uncheck send me SMS notifications | Unchecked box without delay | Box unchecked swiftly | Passed |
| HP4m | High | Click Enter Queue | Pop up indicating incorrect cellphone number format appears | Pop up appeared to re-enter correct format for cellphone number | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.17 VIP login (mobile view) – login with correct password with SMS notifications and prioritize me enabled | | | | | |
| VL1m | Med | Log in with given credentials | Log in successful, page redirected to queue entry | Log in successful and page redirected to queue entry | Passed |
| VL2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| VL3m | Low | Check send SMS notificaitons | Checked box without delay | Box checked swiftly | Passed |
| VL4m | Med | Check Prioritize me box | Checked box without delay | Box checked upon loading of site | Passed |
| VL5m | High | Click Enter Queue | Received SMS notification about queue status | SMS notification received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.18 VIP login (mobile view) – login with correct password with SMS notifications disabled | | | | | |
| VL1m | Med | Log in with given credentials | Log in successful, page redirected to queue entry | Log in successful and page redirected to queue entry | Passed |
| VL2m | Low | Select option from dropdown menu | Select option from dropdown menu without delay | Option selected swiftly | Passed |
| VL3m | Low | Uncheck send SMS notificaitons | Unchecked box without delay | Box unchecked swiftly | Passed |
| VL4m | High | Click Enter Queue | Receive queue number | Queue number received | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.19 VIP login (mobile view) – login with incorrect password | | | | | |
| VL1m | High | Log in with incorrect credentials | Pop up of incorrect password appears | Pop up appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.20 Real-time queue (mobile view) | | | | | |
| RTQ1m | Med | View page | Paged viewed with continuous updates | Page loaded with continuous updates | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.21 Controller (with active queue) – view queue table | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover mouse on view tables | Hovered mouse over view tables | Hovered mouse over view tables | Passed |
| C3 | High | Hover and click view queue | Hovered and clicked view queue | Hovered and clicked view queue | Passed |
| C4 | High | View queue | View database popped up | View database popped up | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.22 Controller (with active queue) – view records history | | | | | |
| C1 | Low | Click Database | Menu clicked | Menu clicked | Passed |
| C2 | Low | Hover mouse on view tables | Selected connect to database | Selected connect to database | Passed |
| C3 | High | Hover and click view records history | Connect clicked | Database connection successful | Passed |
| C4 | High | View queue | View database popped up | View database popped up | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.23 Controller (with active queue) – view VIP list | | | | | |
| C1 | Low | Click Database | Menu clicked | Menu clicked | Passed |
| C2 | Low | Select Connect to database | Selected connect to database | Selected connect to database | Passed |
| C3 | High | Click connect | Connect clicked | Database connection successful | Passed |
| C4 | High | View queue | View database popped up | View database popped up | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.24 Controller (with active queue) – add a VIP with correct number format and confirmation password | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Add | Add hovered | Add hovered | Passed |
| C3 | High | Hover and click on add a VIP | Hovered and clicked on add a VIP | Hovered and clicked on add a VIP | Passed |
| C4 | Med | Enter name | Name entered | Name entered | Passed |
| C5 | Med | Enter mobile number with correct format | Mobile number entered with correct format | Mobile number entered with correct format | Passed |
| C6 | Med | Enter username | Entered username | Username entered | Passed |
| C7 | Med | Enter password | Entered password | Password entered | Passed |
| C8 | Med | Confirm Password | Entered confirm password | Password re-entered | Passed |
| C9 | High | Click OK | VIP record added to list | VIP record added to list | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.25 Controller (with active queue) – add a VIP with incorrect mobile number and correct confirmation password | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Add | Add hovered | Add hovered | Passed |
| C3 | High | Hover and click on add a VIP | Hovered and clicked on add a VIP | Hovered and clicked on add a VIP | Passed |
| C4 | Med | Enter name | Name entered | Name entered | Passed |
| C5 | Med | Enter mobile number with incorrect format | Mobile number entered with incorrect format | Mobile number entered with incorrect format | Passed |
| C6 | Med | Enter username | Entered username | Username entered | Passed |
| C7 | Med | Enter password | Entered password | Password entered | Passed |
| C8 | Med | Confirm Password | Entered confirm password | Password re-entered | Passed |
| C9 | High | Click OK | Warning pop up appears indicating incorrect format with mobile number | Pop up indicating incorrect mobile number format appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.26 Controller (with active queue) – add a VIP with correct mobile number and incorrect confirmation password | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Add | Add hovered | Add hovered | Passed |
| C3 | High | Hover and click on add a VIP | Hovered and clicked on add a VIP | Hovered and clicked on add a VIP | Passed |
| C4 | Med | Enter name | Name entered | Name entered | Passed |
| C5 | Med | Enter mobile number with correct format | Mobile number entered with correct format | Mobile number entered with correct format | Passed |
| C6 | Med | Enter username | Entered username | Username entered | Passed |
| C7 | Med | Enter password | Entered password | Password entered | Passed |
| C8 | Med | Confirm Password entered is different | Entered confirm password differently | Password re-entered differently | Passed |
| C9 | High | Click OK | Pop up indicating wrong confirm password appears | Pop up indicating wrong confirm password appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.27 Controller (with active queue) – add a VIP with incorrect mobile number and incorrect confirmation password | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Add | Add hovered | Add hovered | Passed |
| C3 | High | Hover and click on add a VIP | Hovered and clicked on add a VIP | Hovered and clicked on add a VIP | Passed |
| C4 | Med | Enter name | Name entered | Name entered | Passed |
| C5 | Med | Enter mobile number with incorrect format | Mobile number entered with incorrect format | Mobile number entered with incorrect format | Passed |
| C6 | Med | Enter username | Entered username | Username entered | Passed |
| C7 | Med | Enter password | Entered password | Password entered | Passed |
| C8 | Med | Confirm Password entered is different | Entered confirm password differently | Password re-entered differently | Passed |
| C9 | High | Click OK | Pop up indicating wrong confirm password appears and incorrect format with mobile number | Pop up indicating wrong confirm password appears and incorrect format with mobile number | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.28 Controller (with active queue) – add a transaction | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Add | Add hovered | Add hovered | Passed |
| C3 | Med | Hover and click on add a transaction | Hovered and clicked on add a transaction | Hovered and clicked on add a transaction | Passed |
| C4 | Med | Input transaction name | Transaction name entered | Entered transaction name | Passed |
|  | High | Click OK | Transaction added | Transaction added | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.29 Controller (with active queue) – edit VIP record | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Edit/Delete | Edit/Delete hovered | Edit/Delete hovered | Passed |
| C3 | High | Hover and click on Edit/Delete VIP record | Hovered and clicked on add Edit/Delete VIP record | Hovered and clicked on add Edit/Delete VIP record | Passed |
| C4 | Med | Select username | Username selected | Username selected | Passed |
| C5 | Med | Edit any field | Field edited | Field edited | Passed |
| C6 | High | Click Edit record | Pop up indicating successful editing appears | Pop up indicating successful editing appears | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.30 Controller (with active queue) – delete VIP record | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Edit/Delete | Edit/Delete hovered | Edit/Delete hovered | Passed |
| C3 | High | Hover and click on Edit/Delete VIP record | Hovered and clicked on add Edit/Delete VIP record | Hovered and clicked on add Edit/Delete VIP record | Passed |
| C4 | Med | Select username | Username selected | Username selected | Passed |
| C5 | High | Click Delete record | Record deleted | Record deleted | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.31 Controller (with active queue) – edit transaction | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Edit/Delete | Edit/Delete hovered | Edit/Delete hovered | Passed |
| C3 | High | Hover and click on Edit/Delete transaction | Hovered and clicked on add Edit/Delete transaction | Hovered and clicked on add Edit/Delete transaction | Passed |
| C4 | Low | Select transaction | Transaction selected | Transaction selected | Passed |
| C5 | Low | Click edit transaction | Edit transaction clicked | Edit transaction clicked | Passed |
| C6 | Low | Type new transaction name | Entered new transaction name | Entered new transaction name | Passed |
| C7 | Med | Click OK | Pop up indicating successful editing appeared | Pop up indicating successful editing appeared | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.32 Controller (with active queue) – delete transaction | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on Edit/Delete | Edit/Delete hovered | Edit/Delete hovered | Passed |
| C3 | High | Hover and click on Edit/Delete transaction | Hovered and clicked on add Edit/Delete transaction | Hovered and clicked on add Edit/Delete transaction | Passed |
| C4 | Low | Select transaction | Transaction selected | Transaction selected | Passed |
| C5 | Low | Click delete transaction | Edit transaction clicked | Edit transaction clicked | Passed |
| C6 | Med | Click Yes | Transaction deleted | Transaction deleted | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.33 Controller (with active queue) – cleanup queue | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on clear database | Hovered on clear database | Hovered on clear database | Passed |
| C3 | High | Hover and click cleanup queue | Pop up warning indicating potential data loss warning appeared | Pop up warning indicating potential data loss warning appeared | Passed |
| C4 | Med | Click Yes | Queue cleanup successful | Queue cleanup successful | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.34 Controller (with active queue) – clear queue | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on clear database | Hovered on clear database | Hovered on clear database | Passed |
| C3 | High | Hover and click cleanup queue | Pop up warning indicating potential data loss warning appeared | Pop up warning indicating potential data loss warning appeared | Passed |
| C4 | Med | Click Yes | Clear queue successful | Clear queue successful | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.35 Controller (with active queue) – clear records history | | | | | |
| C1 | Low | Click Database | Database clicked | Database clicked | Passed |
| C2 | Low | Hover on clear database | Hovered on clear database | Hovered on clear database | Passed |
| C3 | High | Hover and click cleanup queue | Pop up warning indicating potential data loss warning appeared | Pop up warning indicating potential data loss warning appeared | Passed |
| C4 | Med | Click Yes | Clear queue successful | Clear queue successful | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.36 Controller (with active queue) – launch display | | | | | |
| C1 | Low | Click Window | Window clicked | Window clicked | Passed |
| C2 | Low | Select Launch Display | Display launched | Display launched | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.37 Controller (with active queue) – fullscreen display | | | | | |
| C1 | Low | Click Window | Window clicked | Window clicked | Passed |
| C2 | Low | Select Fullscreen Display | Display launched in fullscreen | Display launched in fullscreen | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.38 Controller (with active queue) – restore window controls | | | | | |
| C1 | Low | Click Window | Window clicked | Window clicked | Passed |
| C2 | Low | Select Restore window controls | Display launched | Display launched | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.39 Controller (with active queue) – close display | | | | | |
| C1 | Low | Click Window | Window clicked | Window clicked | Passed |
| C2 | Low | Select Close display | Display closed | Display closed | Passed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.40 Controller (with active queue) – call next | | | | | |
| C1 | Low | Click Call next | Next in line is called | Next in line is called | Passed |

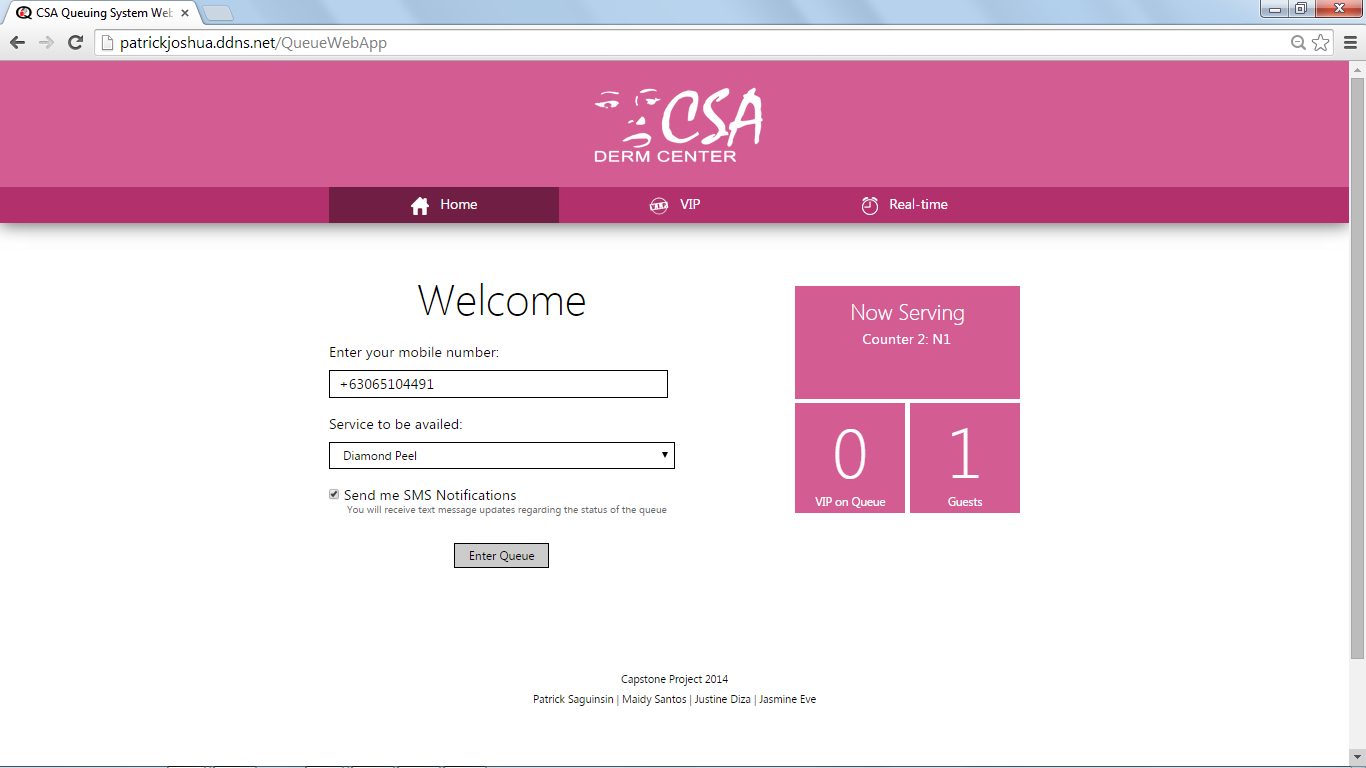
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Priority | Scenario (Steps to Execute) | Expected Result | Actual Result | Remarks (Passed/Failed) |
| Table 4.3.41 Controller (with active queue) – call again | | | | | |
| C1 | Low | Click Call again | Next in line is called again | Next in line is called again | Passed |

### 4.4 Description of the Prototype

The system is a web based queuing management system applicable for any businesses with problems on handling queues. The web application consists of three main menus: home, VIP, and real-time. The home tab displays the main page of the system where regular clients enter their mobile number and select their desired transaction. The VIP tab of the web application is for the preferred clients of the business. It displays the log in page which consists of VIP client’s username and password. They have to use this part of the web application for them be prioritized. The real-time menu is used by any clients to monitor the queue status online. Additionally, the system consists of a controller which is used by authorized personnel to efficiently and effectively navigate the system including all of its features. Also, monitoring queue status is not limited online, it has a display monitor which shows the number currently served and is placed on the establishment’s lounge.

Moreover, the system has the ability to send SMS to clients for the confirmation of their request and for updates regarding the queue status. It also provides printed tickets to clients who do not have a smartphone thus, processed their requests on the actual business site. For defense proper and testing purposes, the system is implemented for CSA Derm Centre, a full service of Cosmetic Dermatology, Anti-Aging Medicine, Slimming, Laser and Spa owned by Dr. Contessa Salvador-Alapag with the main clinic located at 2nd Flr. Puregold Strip Mall Bldg. Mac-Arthur Highway, Mabalacat City, Pampanga since 2010 as of 2014). The prototype were created and tested on the team’s personal devices.

The screenshots found after this statement shows how the prototype looks like.

  
Figure 4.4.1: Web App Home Page

The welcome page of the web app shows a quick summarized view of the system capabilities. Normal clients are the target users of the page, where they can input their mobile number, their choice of transaction or service, and whether to avail the SMS Notification feature. Quick glance-able tiles located on the right side of the page shows the queue status that is updated at the time of loading the page.

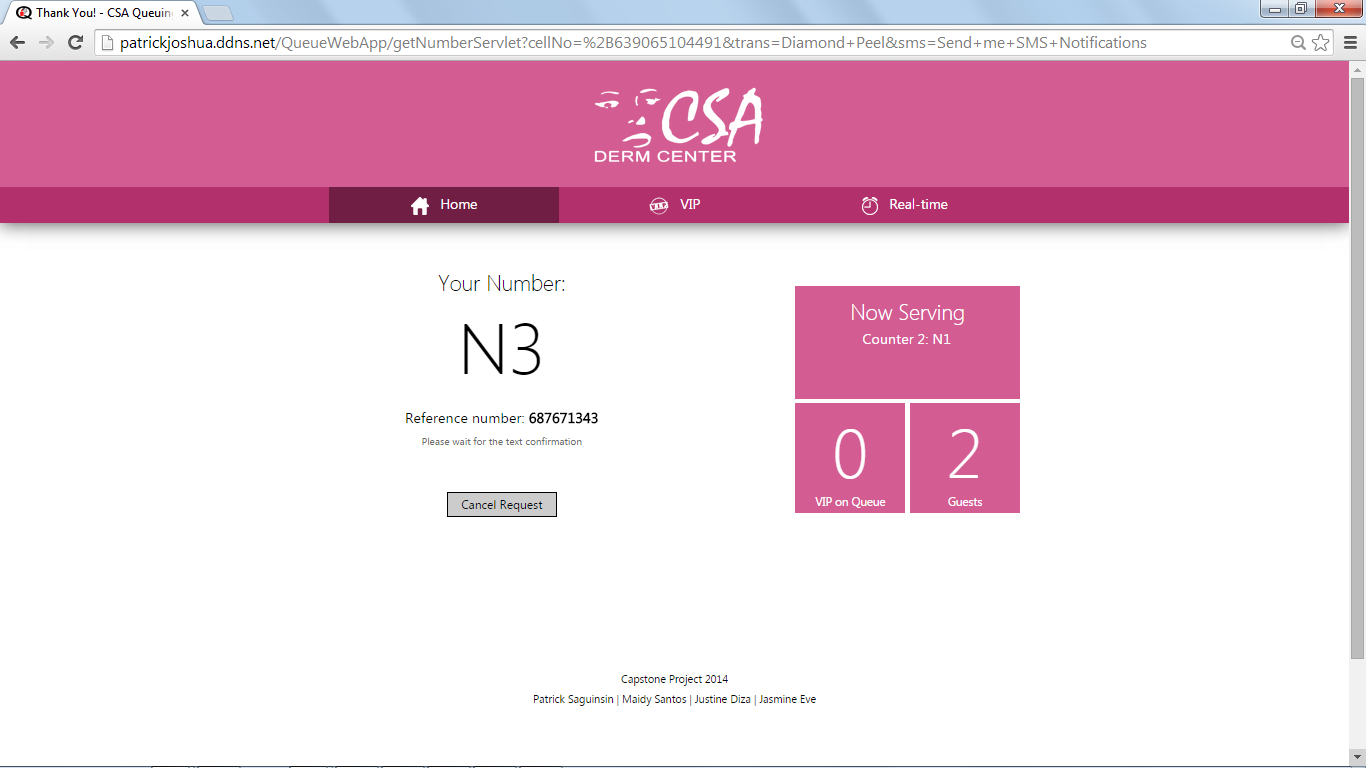


Figure 4.4.2: Web page that display the ticket number after requesting

Ticket number is displayed clearly and distinctively among the rest for a better clarity of the user. Reference numbers are also displayed, as well as the message that informs the client to wait for the text confirmation

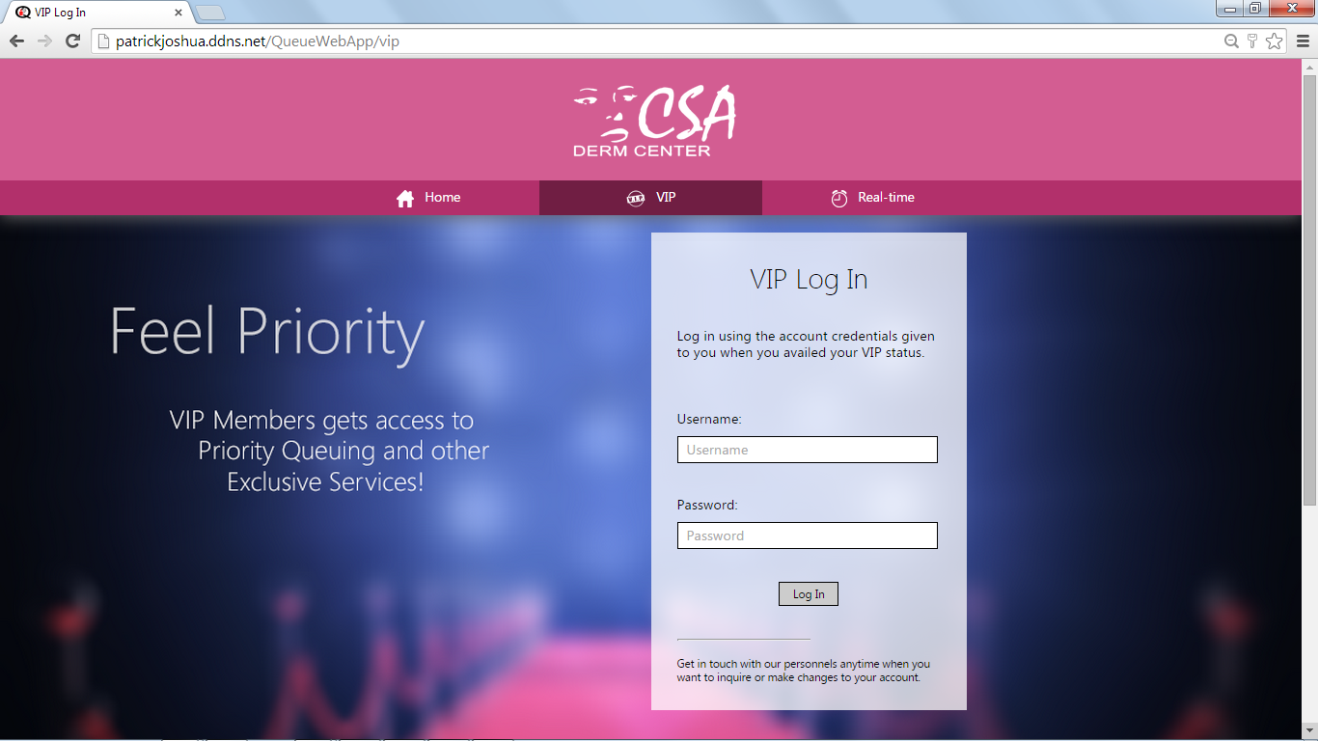


Figure 4.4.3: VIP Login page

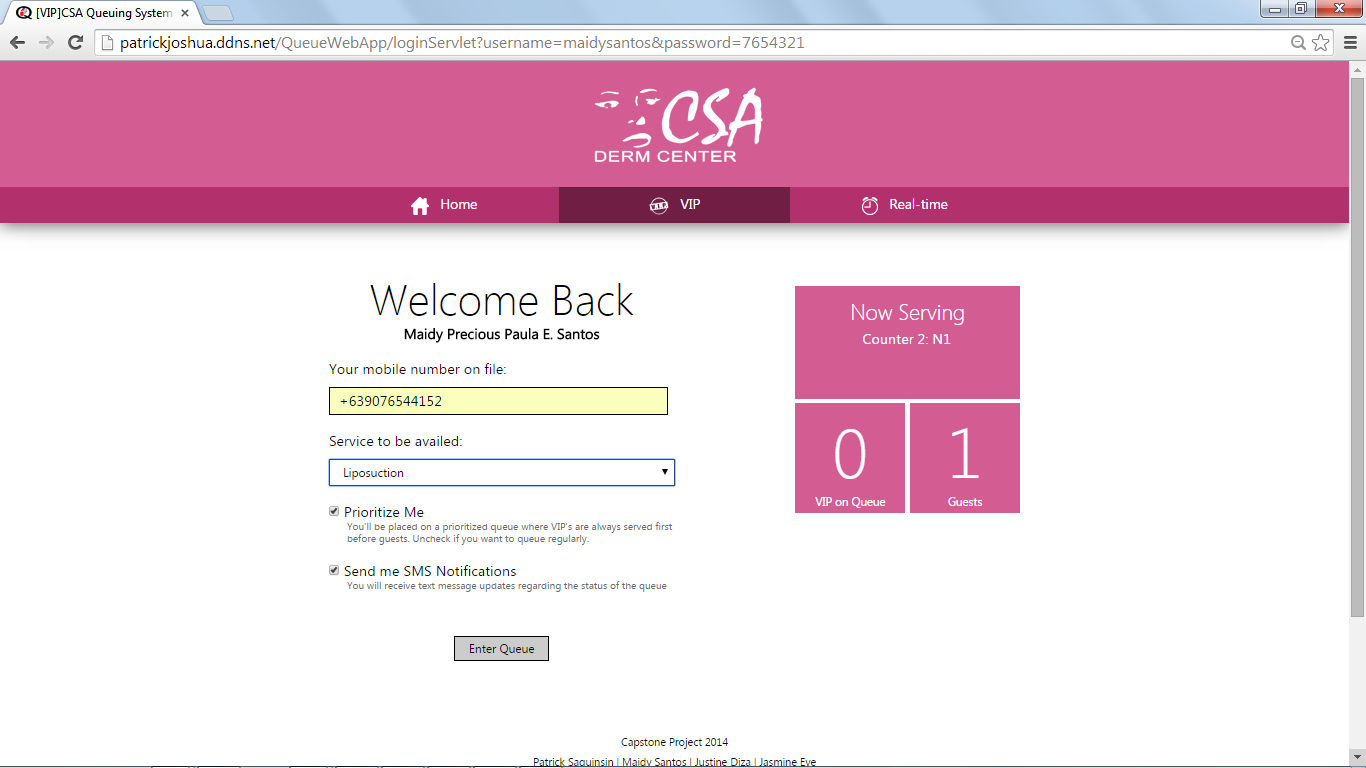


Figure 4.4.4: Web page after login

Figure 4.4.4 shows the web page that displays the mobile number of the client from file, the database. It lets the client change his/her mobile number if he/she currently doesn’t have the phone on record and uses other phone. A familiar “Service to be availed” drop down box lists the available services of the organization. A checkbox indicating “Prioritize Me” gives the client an option to avail the VIP service, or choose to queue as a normal client for courtesy purposes. Finally, the web app also gives option for the client to have SMS Notifications.

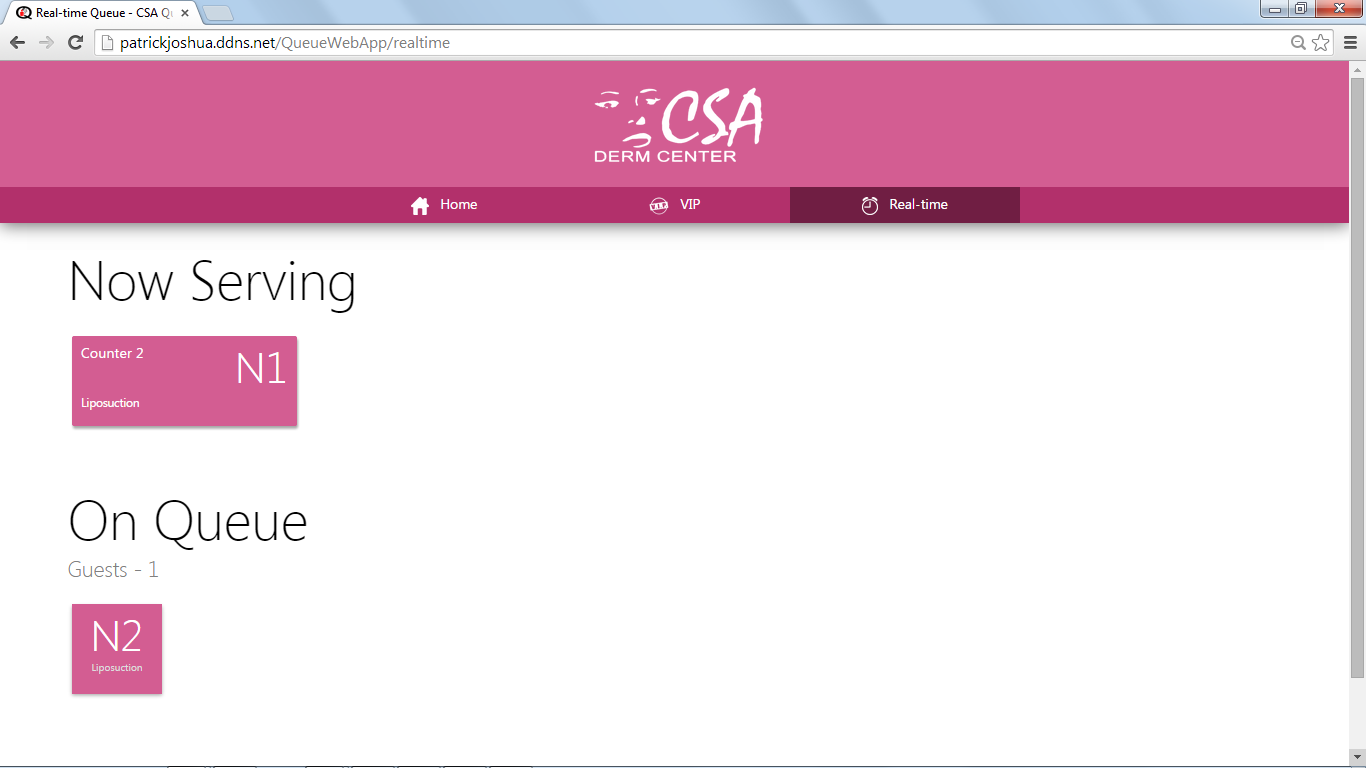


Figure 4.4.1: Real-time Queue Status Page

This figure shows the most dynamic part of the web app, timely updating the information displayed so that clients need not be constrained inside the waiting lounge. Now Serving area displays the number of active counters based on the number of boxes. On Queue section gives a graphical list of the people on queue, and categorizes it based on their priority status. The page also briefly shows the number of persons on queue beside the labels.

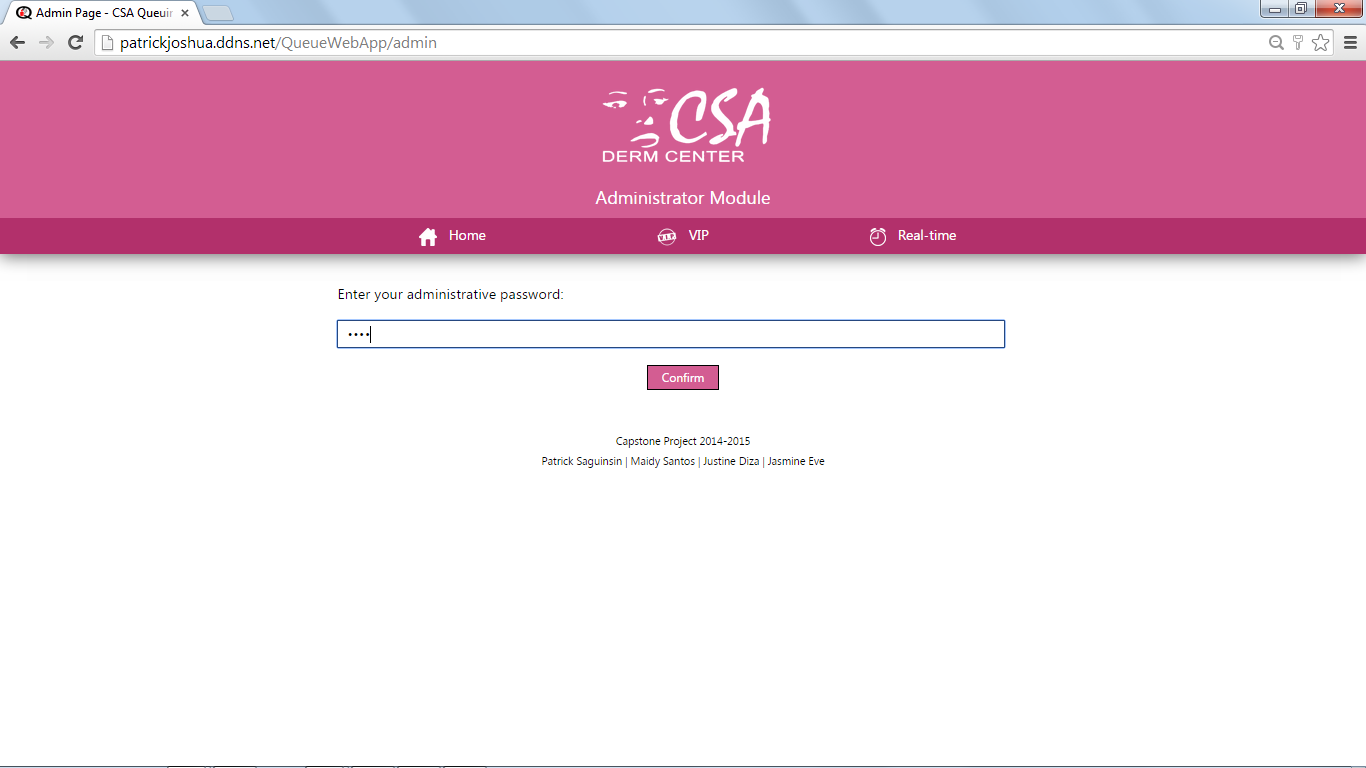
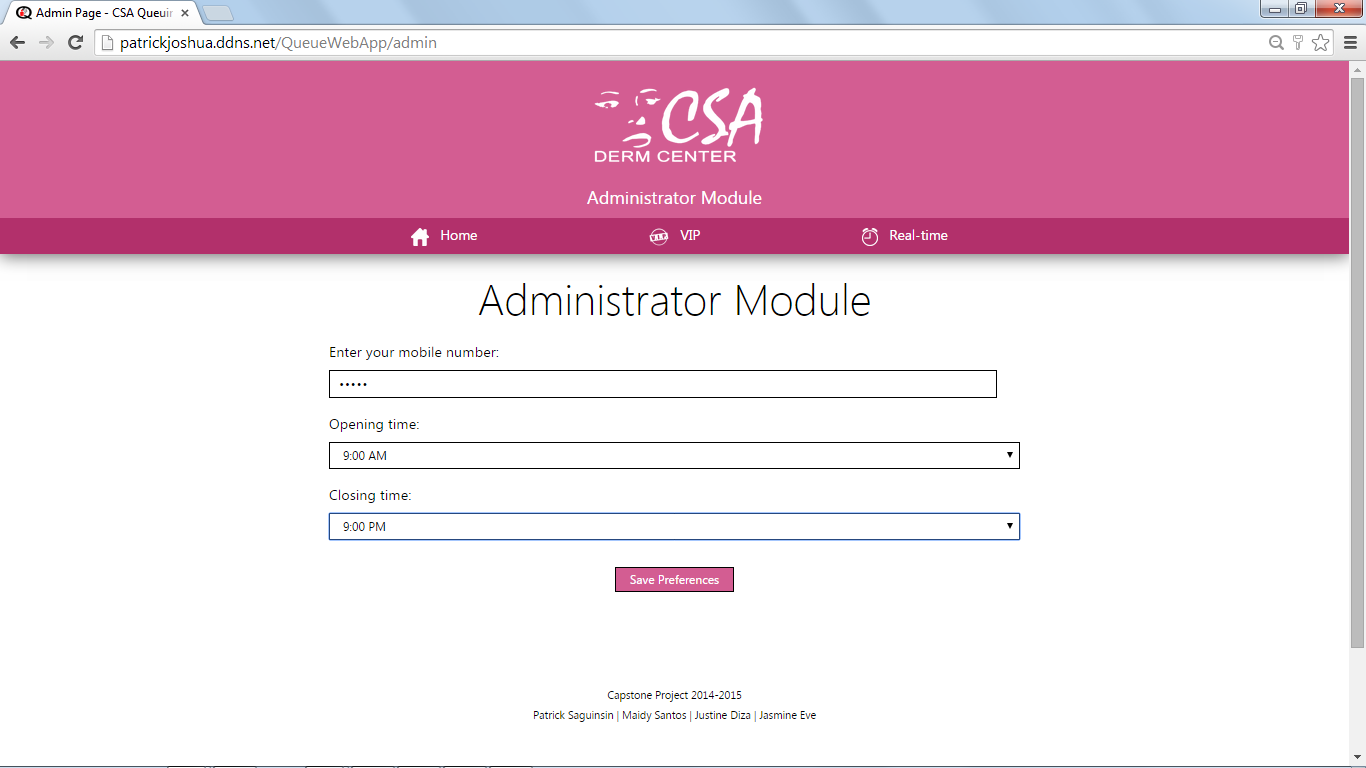


Figure 4.4.6: Administrator Module Security Feature

The administrator module, which is secretly located on a URL that’s not displayed on any page of the web app, will restrict access through an administrative password for authorized users only.

  
Figure 4.4.7: Administrator Module Proper

Upon password verification, the web app briefly changes into the administrator module proper, where the administrator can change his/her password, as well as change the opening and closing time preferences of the web app. Note that the web app dynamically displays “Clinic is now closed” on the real-time modules where applicable if it detects that system is inactive, and through the use of system time, verifies that the store won’t call a number anymore.

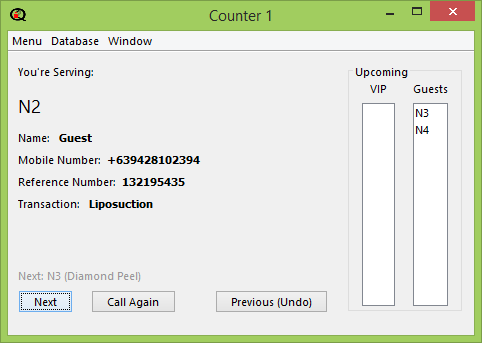


Figure 4.4.8: Controller

The Controller can be implemented in different counters depending on the availability of service counters of the client. It has a Menu for a list of options to provide the preferences of the assigned administrator of the counter, a Database to view the records of the queue as seen on *Figure 4.4.8.1* and *Figure 4.4.8.2*, and a Window to handle the display in ways like opening, closing, minimizing, and maximizing the display window. The controller shows the current queue number being served, the name of the person in the queue if he/she is a VIP member, his/her mobile number, reference number and the transaction type. The queue can be controlled by calling the next person in the queue with the “Next” button, and a “Call Again” button will be active once someone is already added into the queue to send alert to the lounge. The queue for upcoming VIP and Guests customers can also be viewed on the controller window.

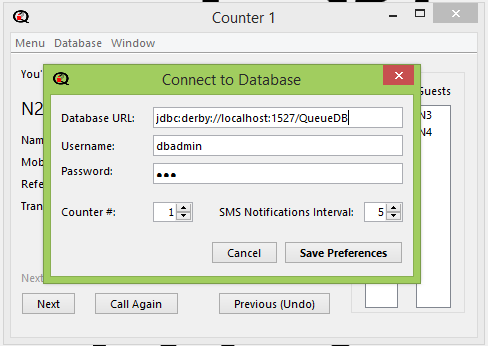


Figure 4.4.8.1: Connecting to System Database

*Figure 4.4.8.1* shows the window that will pop up once the Database option is accessed. Fields for the Database URL, the administrator’s Username and Password, The Counter number and the SMS Notifications Interval will be available for fill-up to save the preferred references.

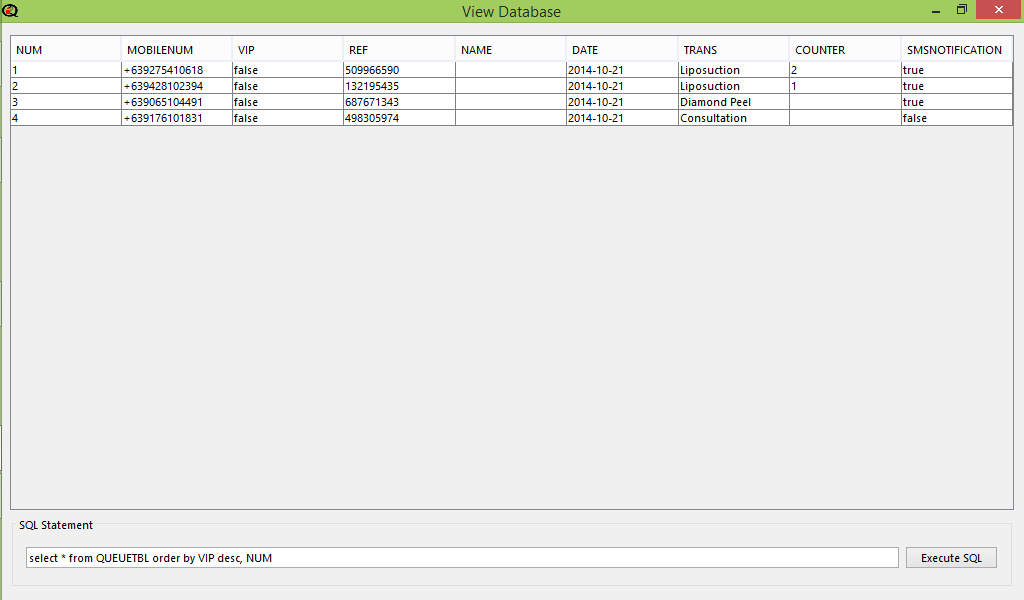


Figure 4.4.8.2: Database Display

In *Figure 4.4.8.2*, the queuing system database will be displayed once it is accessed in the controller window. The database includes the queue number, the mobile number entered by the user, the VIP status of the user, the name (if available, on VIP members), the date, counter number and if the user opted for SMS Notification. SQL statements are also available at the bottom of the window to select and modify the database table.



Figure 4.4.9: Display

And lastly, Figure 4.4.9 shows the main screen that will display queue number of the client who is currently served per counter. There are no controls available for this screen, it merely displays what the administrator would like to project, and it is fully dependent on the administrator’s commands.

## Chapter 5 Recommendations

Even though the Queuing Management System with SMS Notification and Prioritization Web App is functioning well as it is designed to be, there are still more room for improvements that would make it better. The proponents of the project recommend the following for future tests and developments:

* In the near future where almost everyone; young and old, owns not only a cellular phone but a smartphone, the future developers can come up with a Mobile Application version of the project.
* For the improvement of customer service and interaction, integration of SMS conversation is a possibility; for the system to also allow SMS reception and not only SMS transmission.
* Developers can use or provide a faster SMS Gateway, and better if it will be implemented in a lower production cost.
* Implementation of voice and sound technology for input can be optimized for a hands-free entry in the queue
* Implementation of fingerprint scanner can also be used for VIP login instead manually typing the username and password in every session
* Hand gestures and similar technology, most common smartphones, can also be used for manual queuing for a faster entry in the queue
* Developers can use or provide an eco-friendly queue confirmation that does not include printing and/or physical queue numbers
* Effectively applying an accurate computation of waiting time can also be added given that the future developers discovered an efficient and accurate algorithm to minimize the use of time approximation per queue service

## Bibliography

Barclay J. L., Gibson P. H., Lewis A., Wilson C., Affolter J. T., Patel J. C., Scott N. W...Broadhurst P. A. (2013). Adverse psychological effects and costs associated with waiting for radiofrequency ablation. *Wiley Periodicals, Inc., 2013,* 36(6), 719-26. doi: 10.1111/pace.12099

Batt, R. J. & Terwiesch, C. (2013). Waiting Patiently: An Empirical Study of Queue Abandonment in an Emergency Department. The Wharton School, University of Pennsylvania, Philadelphia.

Chowdhury, M. (2013). Queuing Theory Model Used to Solve the Waiting Line of a Bank –A Study on Islami Bank Bangladesh Limited, Chawkbazar Branch, Chittagong. Department of Business Administration, International Islamic University, Chittagong, Bangladesh.

Evans, P. (2013). Philippines - Telecoms, Mobile, Broadband and Forecasts. *Bubble Comm.* Retrieved from: http://www.budde.com.au/Research/Philippines-Telecoms-Mobile-Broadband-and-Forecasts.html

IPFrontline Authors. *OpenText Launches Project Red Oxygen's Process Suite.* Retrieved from http://www.ipfrontline.com/depts/printabletemplate.aspx?id=52033

General Microsystems Sdn Bhd. (n.d.). *Queue Management.* Retrieved from <http://www.gms.com.my/GMS_Home/concept.htm>

Grabis, J. (2013). *Perspectives in Business Informatics Research.* Berlin Heidelberg: Springer.

Gupta, D. (2013). *Handbook of Healthcare Operations Management: Methods and Applications*. New York: Springer.

Hidalgo, V.B. (2011, September 04). BPI rolls out new service. *Inquirer.net.* Retrieved from <http://business.inquirer.net/16795/bpi-rolls-out-new-service>

International Business Machines Corporation. (2012). *SMS notifications.* Retrieved from http://public.dhe.ibm.com/software/mobile-solutions/worklight/docs/v600/09\_05\_SMS\_notifications.pdf

Kokkinou, A., & Cranage, D. (2012). Using self-service technology to reduce customer waiting times. *International Journal of Hospitality Management, 33, 435-445.*

La Torre, S., Madhusudan, P. & Parlato, G. (2008). Context-Bounded Analysis of Concurrent Queue Systems. In Ramakrishnan, C. R. & Rehof, J. (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems* (pp. 299-314). Berlin Heidelberg: Springer.

Lesonsky, R. (2012, June 11). Long Wait Times Harming Retail Experience. *The Huffington Post.* Retrieved from http://www.huffingtonpost.com/2012/06/11/survey-says-long-wait-times-harm-retail\_n\_1577898.html

L. Lakatos et al. (2013). *Introduction to Queueing Systems with Telecommunication Applications.* US: Springer.

Maister, D. (2005). *The Psychology of Waiting Lines.* Retrieved from <http://davidmaister.com/articles/the-psychology-of-waiting-lines/>

Perkins, R. (2012). *Why Waiting is Torture*. Retrieved from <http://www.templealiyah.com/sites/default/files/uploaded_documents/erev_rh_2012_.pdf>

RSI Concepts FZE. (n.d.). *Why Queue Management System.* Retrieved from http://www.rsiconcepts.com/why-queue-management-system.php

Sheahan, T. et al (2013). *Red Oxygen: Crunchbase Review.* Retrieved from http://www.crunchbase.com/organization/red-oxygen#sthash.8pM2yv8r.dpuf

Stone, A. (2012, August 18). Why Waiting Is Torture. *The New York Times*. Retrieved from <http://www.nytimes.com/2012/08/19/opinion/sunday/why-waiting-in-line-is-torture.html?pagewanted=all&_r=1&>

Suarez, KD (2012, July 18). ‘Mobile revolution’ changing the world. *Rappler.* Retrieved from <http://www.rappler.com/life-and-style/technology/8786-it-s-a-mobile-world>

SWNS Reporter (2012, September 27). Wait in line: Nearly a year of our life is spent in QUEUES. *SWNS.* Retrieved from <http://swns.com/news/wait-in-line-nearly-a-year-of-our-life-is-spent-in-queues-24514/>

The Nielsen Company. (2014). *The Asian Mobile Consumer Decode.* Retrieved from http://www.nielsen.com/content/corporate/ph/en/insights/news/2014/asian-mobile-consumers.html

Thomopoulos, N. (2012). *Fundamentals of Queuing Systems: Statistical Methods for Analyzing Queuing Models*. New York: Springer US.