Software Requirement Specification

for

Merchant Queueing System (MQS)

Version 1.0

Prepared by Ginanjar Fahrul M

February, 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason** | **Version** |
| Gin | 2/10/2014 | Initial requirement draft | 1.0 |

# Introduction

## Purpose

The purpose of this document is to enumerate and describe requirements specific to the Merchant Queueing System (MQS) application. This Software Requirement Specification (SRS) defines the requirements for the entire first build of the system.

## Product Scope

The MQS application is a mobile application that our client (merchant) will use in order to simplify the customer manual queue service. MQS will have a centralized data storage of all client who come to a specific merchant and speed up and make the queueing more efficient.

## References

This document references the following documents:

* <http://www.ticktok.com.sg>
* Software Requirements Specification for MCQA, UVA-Wise Capstone

# Overall Description

## Product Functions

The product shall support:

* Create an automated queueing system for merchant’s customers.
* Sending and receiving information about customer’s habits.
* Sending all necessary queue status information from merchant to customer.

## User Classes and Characteristics

We anticipate trained waiters will use the software on daily basis to assist the customer. Management will use the software to get a frequently report from system. Technical support customer personnel will use the web/mobile client to fix system configuration.

## Operating Environment

The software shall run on any system that the Android mobile platform operates on. The software shall be tested on the Samsung Galaxy Tab 2 or similiar Android Tablet. The software shall utilize communication capabilities through the internet. The software shall run a server which provide back-end functionality with the call/sms capabilities. The client-server architecture must follow the specifics API protocol.

## User Documentation

The final deliverable shall include the software document folder (SDF) which contains tutorials on Git, UML Tools, DroidDraw, and user manuals for both the mobile client and web client.

# Functional Requirements

This sections details the main feature of the system by listing the actions to be taken and defining the requirements for those features.

## Mobile Client Requirements (MCR)

* **MCR-01**: The software shall allow the automatic registration based on customer specific information.
* **MCR-02**: The software shall provide the customer data who has been registered.
* **MCR-03**: The software shall allow the self service function for customer to get the queue number from system.
* **MCR-04**: The software shall automatically show necessary queue status information (queue number, predicted waiting time, etc).
* **MCR-06**: The software shall allow the customer check-in/check-out when the customer’s queue number is called or finished.

## Web Client Requirements (WCR)

* **WCR-01**: The software shall provide the real time updates to the customers for their queue status information (predicted waiting time, queue number to call, etc).
* **WCR-02**: The software shall allow the user to query the information from the application.
* **WCR-03**: The software shall restrict the customer’s session who reach the information from the application.
* **WCR-04**: The software shall be capable of allowing the customer to cancel the reservation automatically (time-out) and manually.
* **WCR-05**: The software shall provide the technical support user to configure the merchant’s profiles and configurations.
* **WCR-06**: The software shall provide the report to the manager.
* **WCR-07**: If the client install mobile app, the software shall provide automatic check-in/check-out based on GPS information.

## Server Database Requirements (SDR)

* **SDR-01**: The software shall contain a design capable of storing all necessary information regarding customers.
* **SDR-02**: The software shall only send data upon request from a client system.
* **SDR-03**: The software shall provide an automatic notification to the customer about their queue status information (by call or SMS).

# Nonfunctional Requirements

## Performance Requirements

* **SQE-1**: The software shall use a rewritable data storage to handle reloads after crashes and will be limited to 20 megabytes in order to reduce space used.
* **SQRE-1**: The software shall have a crash recovery system so that no unsent data will be lost.
* **SQM-1**: The software modules shall be limited to 30 statements.
* **SQF-1**: The software shall implement Live Software Updates.
* **SQU-2**: The software shall resend data every TBD seconds, if no response from server, for a maximum of TBD times.
* **SQE-2**: The software shall operate for a minimum of 4 hour on a standard factory configured mobile platform.

## Safety Requirements

* **SQU-1**: The software shall include an operations manual.

## Security Requirements

* **SQI-1**: The software shall restrict the customer’s session who reach the information from the application.
* **SQI-2**: The software shall have user logins to validate permission levels for merchants.
* **SQIP-2**: The software shall use a TBD API format with a restricted session for client-server communications.

## Software Quality Attributes

* **SQP-1**: The software data shall be accessible on any platform that supports Google Android OS above versions 2.2.
* **SQP-2**: The software shall include a web client.
* **SQC-1**: The software shall follow the Java Coding Style Guide as based on the [http://developers.sun.com/sunstudio/products/archive/whitepapers/java-style.pdf](http://developers.sun.com/sunstudio/products/archive/whitepapers/java-style.pdf%20) site.
* **SQM-2**: The software shall implement standardized error messaging, GUI layouts, and hardware interfacing based on TBD format.

## Business Rules

* **SQIP-1**: The software shall use a simple database management on server/client.
* **SQRE-2**: The software shall be written in Java for mobile client and PHP for web client/API.

# Appendix A: Glossary

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| SRS | Software Requirements Specification |
| GUI | Graphic User Interface |
| API | Application Programming Interface |
| MCQA | Mobile Client Query Application |
| MCR | Mobile Client Requirement |
| WCR | Web Client Requirement |
| SDR | Server/Database Requirement |
| SDF | Software Document Folder |
| SQE | Software Quality Efficiency |
| SQC | Software Quality Correctness |
| SQRE | Software Quality Reliability |
| SQI | Software Quality Reliability |
| SQU | Software Quality Usability |
| SQM | Software Quality Maintainability |
| SQF | Software Quality Flexibility |
| SQT | Software Quality Testability |
| SQP | Software Quality Portability |
| SQIP | Software Quality Interoperability |
| OCR | Optical Character Recognition |
| GPS | Global Positioning System |
| DBMS | Database Management System |

# Appendix B: Indexed Requirements

## Mobile Client Requirements (MCR)

* **MCR-01**: The software shall allow the automatic registration based on customer specific information.
* **MCR-02**: The software shall provide the customer data who has been registered.
* **MCR-03**: The software shall allow the self service function for customer to get the queue number from system.
* **MCR-04**: The software shall automatically show necessary queue status information (queue number, predicted waiting time, etc).
* **MCR-06**: The software shall allow the customer check-in/check-out when the customer’s queue number is called or finished.

## Web Client Requirements (WCR)

* **WCR-01**: The software shall provide the real time updates to the customers for their queue status information (predicted waiting time, queue number to call, etc).
* **WCR-02**: The software shall allow the user to query the information from the application.
* **WCR-03**: The software shall restrict the customer’s session who reach the information from the application.
* **WCR-04**: The software shall be capable of allowing the customer to cancel the reservation automatically (time-out) and manually.
* **WCR-05**: The software shall provide the technical support user to configure the merchant’s profiles and configurations.
* **WCR-06**: The software shall provide the report to the manager.
* **WCR-07**: If the client install mobile app, the client shall provide automatic check-in/check-out based on GPS information.

## Server Database Requirements (SDR)

* **SDR-01**: The software shall contain a design capable of storing all necessary information regarding customers.
* **SDR-02**: The software shall only send data upon request from a client system.
* **SDR-03**: The software shall provide an automatic notification to the customer about their queue status information (by call or SMS).

## Product Operation Requirements

Correctness

* **SQC-1**: The software shall follow the Java Coding Style Guide as based on the http://developers.sun.com/sunstudio/products/archive/whitepapers/java-style.pdf site.

Reliability

* **SQRE-1**: The software shall have a crash recovery system so that no unsent data will be lost.
* **SQRE-2**: The software shall be written in Java for mobile client and PHP for web client and TBD API.

Efficiency

* **SQE-1**: The software shall use a rewritable data storage to handle reloads after crashes and will be limited to 20 megabytes in order to reduce space used.
* **SQE-2**: The software shall operate for a minimum of 4 hour on a standard factory configured mobile platform.

Integrity

* **SQI-1**: The software shall restrict the customer’s session who reach the information from the application.
* **SQI-2**: The software shall have user logins to validate permission levels for merchants.

Usability

* **SQU-1**: The software shall include an operations manual.
* **SQU-2**: The software shall resend data every TBD seconds, if no response from server, for a maximum of TBD times.

## Product Revision Requirements

Maintainability

* **SQM-1**: The software modules shall be limited to 30 statements.
* **SQM-2**: The software shall implement standardized error messaging, GUI layouts, and hardware interfacing based on TBD format.

Flexibility

* **SQF-1**: The software shall implement Live Software Updates.

Testability

* **SQT-1**: The software shall implement diagnostic logging on the server and client applications.

## Product Transition Requirements

Portability

* **SQP-1**: The software data shall be accessible on any platform that supports Google Android OS above versions 2.2.
* **SQP-2**: The software shall include a web client.

Interoperability

* **SQIP-1**: The software shall use a simple database management on server/client.
* **SQIP-2**: The software shall use a TBD API format with a restricted session for client-server communications.

# Appendix C: To Be Determined List

Quick References to TBD Items

* **SQRE-2**: The software shall be written in Java for mobile client and PHP for web client and TBD API.
* **SQU-2**: The software shall resend data every TBD seconds, if no response from server, for a maximum of TBD times.
* **SQM-2**: The software shall implement standardized error messaging, GUI layouts, and hardware interfacing based on TBD format.
* **SQIP-2**: The software shall use a TBD API format with a restricted session for client-server communications.

# 

# 

Software Process Specification

for

Merchant Queueing System (MQS)

Version 1.0

Prepared by Ginanjar Fahrul M

February, 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason** | **Version** |
| Gin | 2/12/2014 | Initial requirement draft | 1.0 |

# 

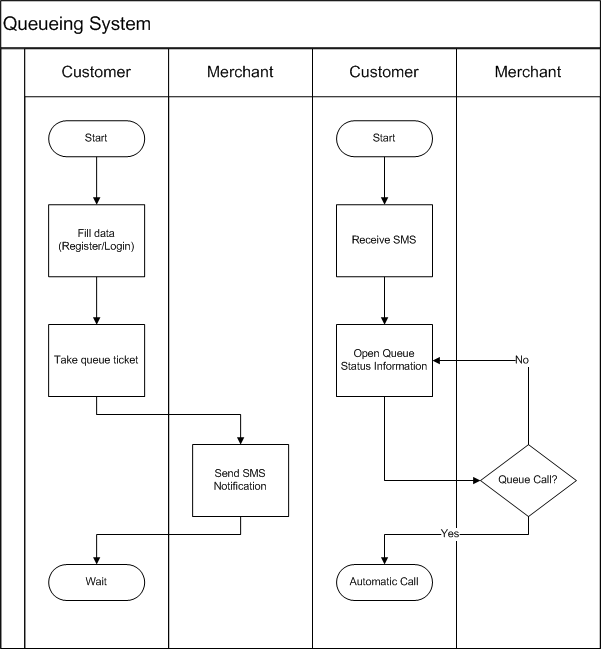
# Introduction

## Purpose

The purpose of this document is to enumerate and describe general business and software process of Merchant Queueing System (MQS). This document will guide the development of MQS.

## General Business Process

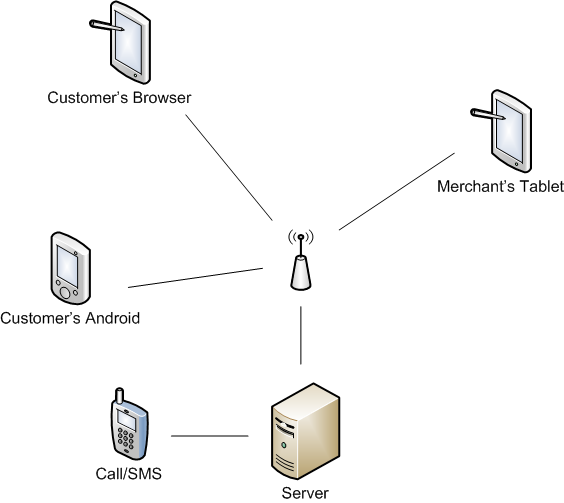
The section here will provide short business process or queueing.



# Proposed System

## System Architecture

Merchant Queueing System can be viewed at Merchant’s Tablet (**U-MT**), Customer’s Browser (**U-CB**), Customer’s Android (**U-CA**), and Customer’s Phone (**U-CP**).



## Device Client Capabilities

* **U-MT**: Android Tablet that MQS app for Merchant installed.
* **U-CB**: Device can open web browser.
* **U-CA**: Android device that MQS app for Customer installed.
* **U-CP**: Any device that has phone call and SMS capabilities.

## Software Features

**Customer**

* Register customer info detail. (**U-CB**, **U-CA**)
* View merchant status (open, closed, queue: full or available). (**U-CB**, **U-CA**)
* Take queue ticket. (**U-CB**, **U-CA**)
* View queue status. (**U-CB**, **U-CA**)
* Cancel queue. (**U-CB**, **U-CA**)
* Receive phone call and SMS notification. (**U-CP**)

**Merchant**

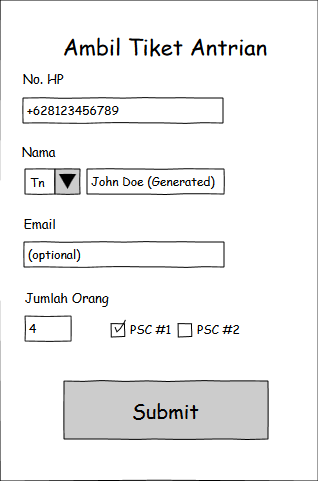
* Register customer info detail. (**U-MT**)
* View dashboard. (**U-MT**)
* Take queue ticket for customer. (**U-MT**)
* View overall queue status. (**U-MT**)
* Cancel customer’s ticket. (**U-MT**)
* Reorder queue. (**U-MT**)

## User Stories

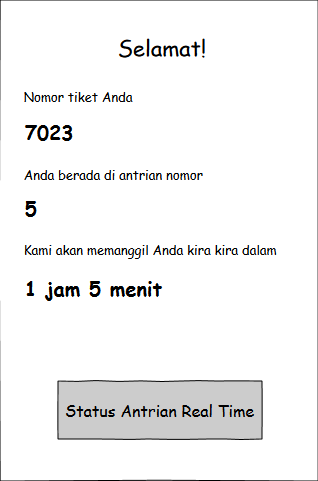
There are three main user stories:

* User Story 1: Customer Press Queue Ticket
* User Story 2: Customer Real Time Update
* User Story 3: Merchant Dashboard

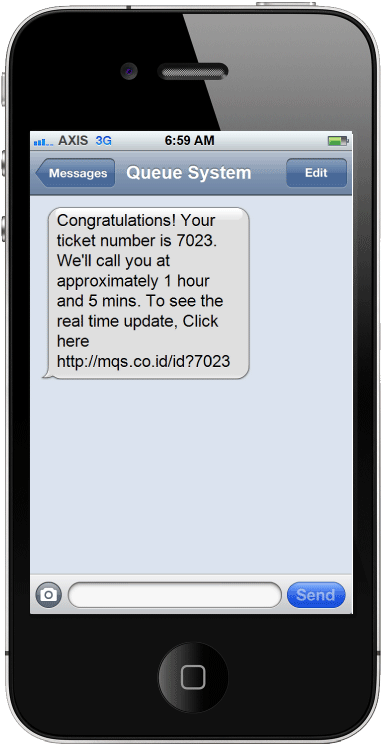
**Customer Press Queue Ticket**



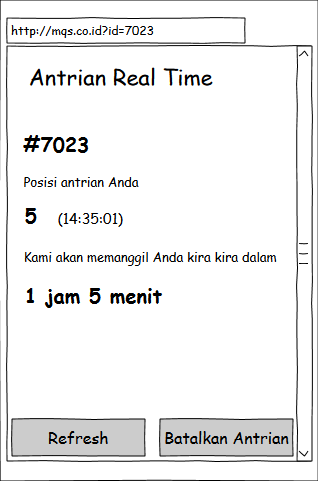
* Customer input
  + phone number
  + name: generated if user is registered
  + number of seat



* Customer receive a queue ticket including
  + ticket number
  + timestamp
  + estimated queue call

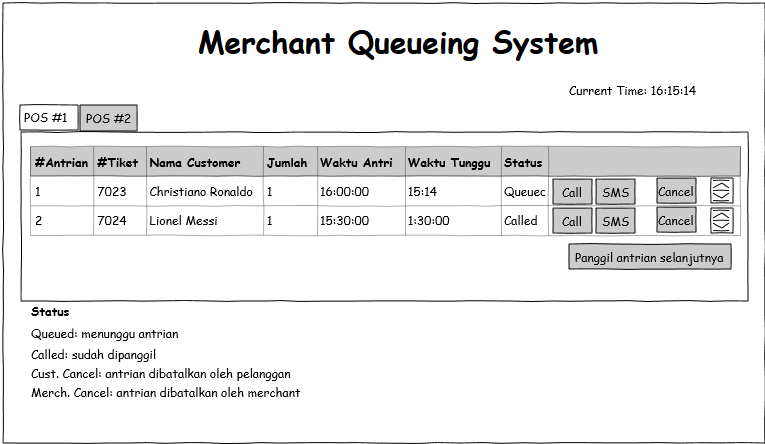


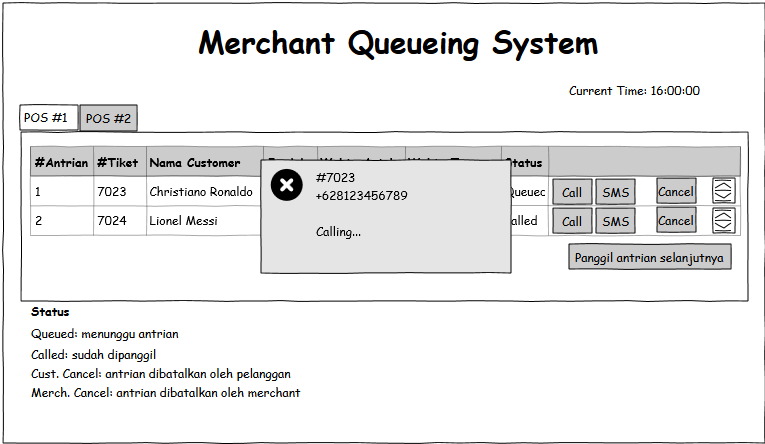
* Customer receive SMS.



**Customer Real Time Update**

* Customer see queue status and estimated time.
* Possibility actions
  + Cancel ticket

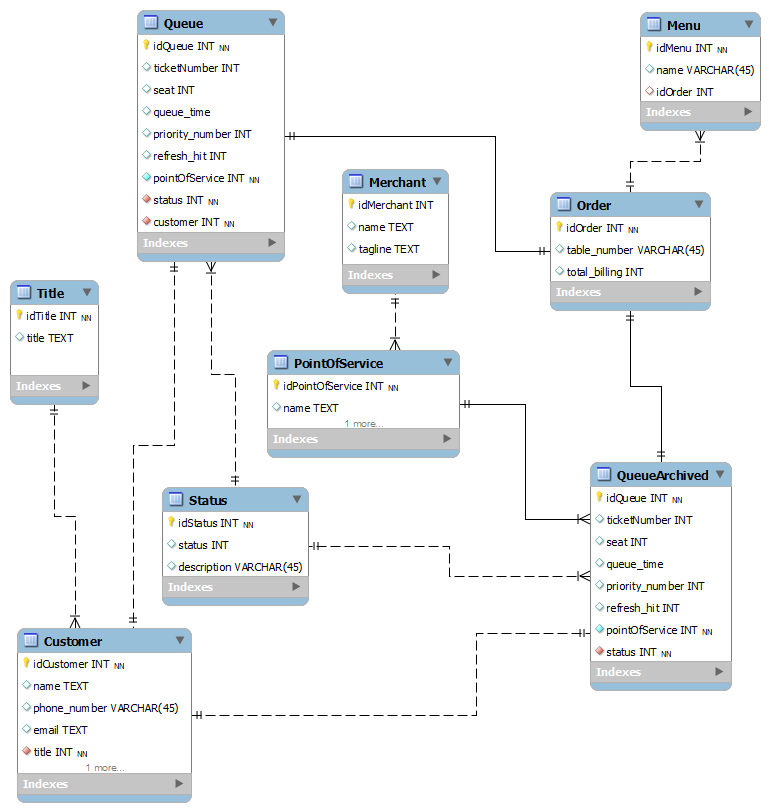




**Merchant Dashboard**

* Merchant see the dashboard.
* Possibility actions
  + Reorder queue
  + Cancel queue
  + Call queue

## Database Design



~\Oo/~