**Design Document**

CIS 4911 – Senior Project

Virtual Queue

**Member:**

Kely Cid

**Instructor**

Masoud Sadjadi

**Mentor**

[Bernard Parenteau](http://spws-dev.cis.fiu.edu/senior-project-website-v4/user/231)

**Date**

October 8th 2014

Copyright © Florida International University 2014. All Rights Reserved. No part of the Virtual Queue Project or documentation may be reproduced or modified without the express consent of Florida International University, School of Computing & Information Sciences. Redistribution or commercial use is strictly prohibited.

**ABSTRACT**

The Design Document gives a better understanding of the Virtual Queue system structure. The reader will be able to capture the idea of how the Virtual Queue system was implemented because the design methodology will be explained, as well as the system architecture and subsystem decomposition, the security and privacy of the system, software and hardware mapping, and persistent data management. Chapter 1 gives basic information about the Virtual Queue (VQ) system, including introduction, problem definition, design methodology used, definitions, acronyms, and overview of the document. Chapter 2 will describe the system decomposition of the VQ by giving an overview of the system, provide a detailed description of the subsystem decomposition, map the hardware and software, identify the persistent data management, and describe security and privacy.

Chapter 3 introduces the detailed design chapter starting with an overview of the behavior and structure of each subsystem, the static and dynamic diagram model, and a description of the code specification. Chapter 4 will have the glossary of terms used in the document, specially the domain specific terms. Chapter 5 contains the appendix of the project with the use case diagrams for the implemented use cases, document class interfaces and diary of meeting and tasks. Finally, Chapter 6 includes any other documents that have been used for reference.

Contents

1. Introduction 5

1.1. Problem definition 5

1.2.   Design methodology used 5

1.3 Terminology - Definitions, acronyms, and abbreviations. 5

1.4.   Overview of document 6

2. System Design (i.e., overall system design) 6

2.1 Overview 7

2.2 Subsystem Decomposition 7

2.3 Hardware-Software Mapping 7

2.4 Persistent Data Management 8

2.5 Security/Privacy 9

3. Detailed Design 10

3.1 Overview 10

3.2 Static models 10

3.3 Dynamic Model 10

3.4 Code Specification 10

4. Glossary 11

5. Appendix 11

5.1 Appendix A - Use case diagram for use cases being implemented 11

5.2 Appendix B - Use cases being implemented (from the RD). 12

5.3 Appendix C – Documented Class Interfaces (code) for the subsystem(s) you will implement and the constraints. 24

5.4 Appendix D – Diary of Meetings 24

6. References 30

# 1. Introduction

The introductory chapter gives some background information about the Virtual Queue system. In the following sections, the problem definition, and scope of the system will be described. Following, the design methodology used is identified. In addition, definitions, acronyms, and abbreviations of terms used in this deliverable will be provided and explained on this chapter. Finally, the chapter will conclude with a brief explanation of what to expect from the following chapters of the document.

## 1.1. Problem definition

When going to a park, or to any other venue that provides multiple recurring rides or events, customers typically wait in line until is time for them to go into the ride or event. This is definitely time consuming, since one could be doing something else like be walking around, buying souvenirs, or food, or going perhaps to another ride or event. By waiting in line, the venue is making money for that specific ride/event, but is losing potential additional sales by customers being in line rather than walking around the venue visiting other areas of the venue.

The creation of the Virtual Queue system is the proposed solution to the problem explained above. The system will provide customers the benefit of enjoying other amenities offered in the venue (including, but not limited to rides, food) instead of waiting in line. In addition, the system will keep information about ride or event time and capacity and allow the customers to sign in to different events or rides. In this way, customers will not have the need to wait for a ride in line to go to another one. Furthermore, customers will be notified as their time on their rides or events approaches. At the same time, the business will take advantage of this system because of the fact that customers will now have the opportunity to either sign in to another ride, or walk around to buy souvenirs, food, or something else offered at the specific place.

## 1.2.   Design methodology used

## 1.3 Terminology - Definitions, acronyms, and abbreviations.

**Definitions**

* **Guest Users**: Anyone who wants to browse through the site and view offered deals before placing an order.
* **Registered Users**: Users that have already created an online account and can place orders and view previous orders they have requested.
* **System:** The system itself.
* **Theme park/Event**: Amusement park with a unifying setting or idea.

**Acronyms**

* **VQ**: Virtual Queue
* **FIU:** Florida International University
* **SCIS:** School of Computing & Information Sciences

**Abbreviations**

As of right now, there are no abbreviations for this project.

## 1.4.   Overview of document

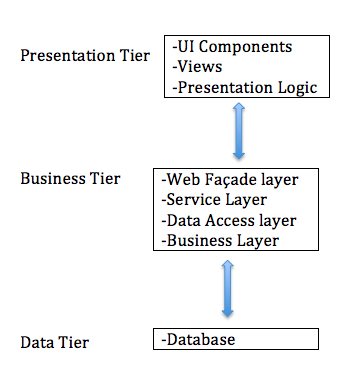
In chapter 1, the main problem is introduced, along with the design methodology used for the project, definitions, acronyms and abbreviations. In chapter 2 the system decomposition of the VQ system will be described, and a detailed description of the subsystem decomposition, map of the hardware and software, identification of the persistent data management, and a description of security and privacy will be provided.

Following, chapter 3 will introduce the detailed design chapter giving an overview of the behavior and structure of each subsystem, the static and dynamic diagram model, and a description of the code specification. Consequently, chapter 4 will have the glossary of terms used in the document, specially the domain specific terms. Chapter 5 will contain the appendix of the project with the use case diagrams for the implemented use cases, document class interfaces and diary of meeting and tasks. Finally, Chapter 6 will include any other documents that have been used for reference.

# 2. System Design (i.e., overall system design)

This chapter will describe the system and subsystem design. It will explain the decomposition of the VQ by giving an overview of the system design architecture. It will provide a detailed description of the subsystem decomposition for each major subsystem. It will cover how the hardware and software are mapped. It will identify the persistent data management that needs to be stored and the structure of the data. Lastly, it will describe security and privacy user authentication processes, encryption of data and all other security parameters being implemented.

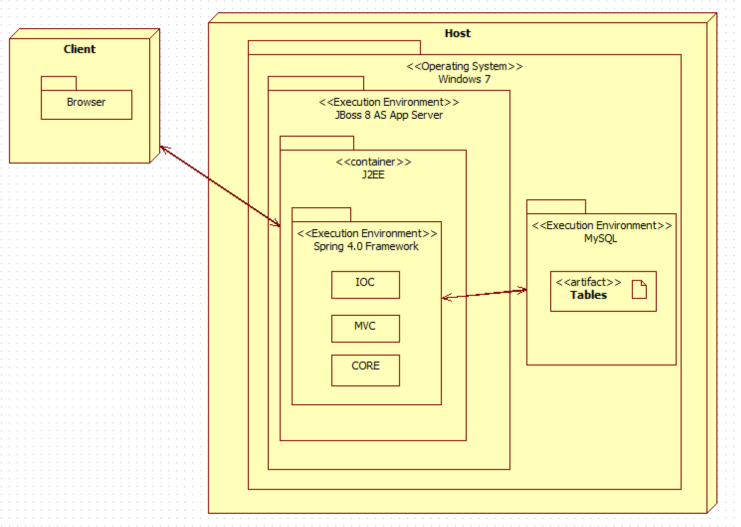
## 2.1 Overview



## 2.2 Subsystem Decomposition

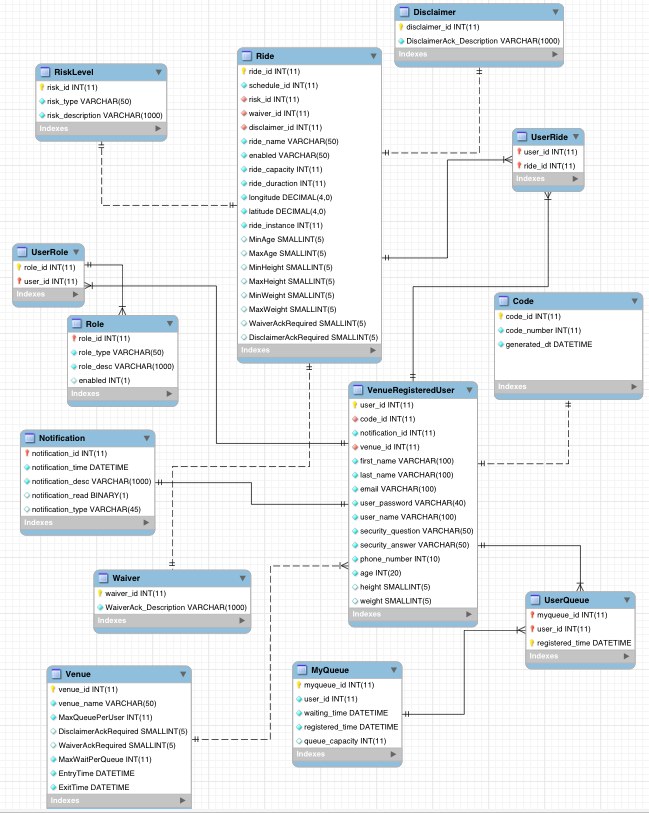
## 2.3 Hardware-Software Mapping

To map the hardware and software used for the Virtual Queue system, a deployment diagram was constructed. It is being hosted on a virtual machine with windows 7 running with JBoss 8 application server and Spring 4.0 framework and MySQL database on the FIU SCIS network. Below is the representation of it.



## 2.4 Persistent Data Management

For the VQ system a database was implemented from scratch, taking into account all the information needed to be stored and support the each functionality. Below is the EER diagram schema with all the tables and relations.



## 2.5 Security/Privacy

For the VQ system all data is password protected. Authentication of each registered user and administrators will be done by checking the database for a registered email, which will be the username as well; also, the table Role and User Role will be checked as well to determine user privileges. Registered users will only have access to their information, and their own data. On the other hand, administrators will have access to all registered users information and results.

# 3. Detailed Design

The detailed design of the VQ system is considered an abstract that translate to source code in a clear way. That is one of the reasons that it has to be detailed and clear. The detailed design chapter will present the system design in a variety of views where each uses a variety of modeling techniques. This chapter is composed for four sections. Section 3.1 is the overview of the chapter; it will give a brief description of the behavior and structure of each subsystem. Section 3.2 contains the static models of each subsystem, and Section 3.3 the dynamic model, which will provide the different diagrams for each subsystem. Section 3.4 delivers the class interfaces and constraints for the main control object in each system.

## 3.1 Overview

## 3.2 Static models

## 3.3 Dynamic Model

## 

## 3.4 Code Specification

# 4. Glossary

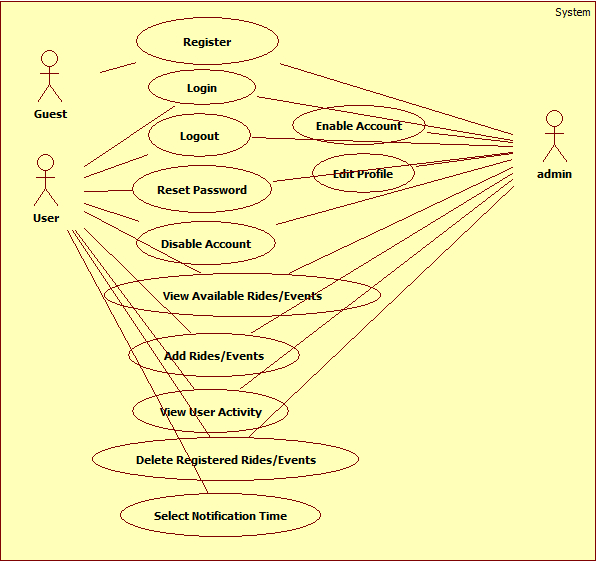
**UML**: Stands for Unified Modeling Language. It is a standardized language that is used to model various things within the field of software engineering.

**Theme park:** An amusement park with a unifying setting or idea.

**Task:** A piece of job that serves as a unit of work.

# 5. Appendix

## 5.1 Appendix A - Use case diagram for use cases being implemented



## 5.2 Appendix B - Use cases being implemented (from the RD).

*Use Case ID:* **VQ01 – User Login**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-conditions:*

1. Web page has been activated
2. Participants must have a web username and password.

*Description:*

1. Use case begins when the user accesses the login option.
2. The user will be prompted with a data entry template for username and password.
3. The user provides a previously registered user name and password.
4. User shall continue request by accessing the Login.
5. If the credentials are valid, the system will log the user into the corresponding page.
6. Use case ends when access is granted to respective user.

*Relevant requirements:* A user will only be allowed into the system if he/she has a valid username and password.

*Post-conditions:*

1. Access is granted.

*Alternative Courses of Action:*

1. In step 2 of Description section there is an option that allows the user to reset the password if he/she forgot. (see use Case **VQ03 – Reset Password**)

*Exceptions:*

1. The login option on the Website is not active.
2. The option to reset password is not active.
3. The database is not active.

*Related Use Cases:* None.

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ02 – User Logout**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-conditions:*

1. Web page has been activated.
2. User must have been previously logged in.

*Description:*

1. Use case begins when the user accesses the logout option.
2. The system logs the user out.
3. Use case ends when the system logs user out and displays the home screen.

*Relevant requirements:* A user will only be logged out if he/she has previously logged in.

*Post-conditions:*

1. User gets successfully logged out of the system.

*Alternative Courses of Action:*

1. The user closes the webpage.

*Exceptions:*

1. The logout option on the webpage is not active.

*Related Use Cases:* None.

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ03 – Reset Password**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-conditions:*

1. User must have previously registered into the system.
2. User must be logged out of the system.

*Description:*

1. Use case begins when the user accesses the option to reset password on the login screen.
2. The system will prompt the user with a data entry template.
3. The user should enter username, which will be their registered email, choose their security question, and write their security answer, new password and password confirmation.
4. The user shall complete the form and submit the changes.
5. If the username is on record, the system will update the password notify the user if the request was submitted successfully. (See Use case **VQ04 – Confirm Identity**).
6. Use case ends when the request is stored in the system.

*Relevant requirements:* A user will only be allowed to reset password if he/she has previously registered.

*Post-conditions:*

1. The password gets successfully changed in the database.

*Alternative Courses of Action:*

1. In step 3 of Description section the username could not be found.
2. In step 3 of Description section the user has the option to cancel the request.
3. In step 3 of Description section the answer to the security question does not match the records.
4. In step 4 of Description section the user has the option to cancel the request.

*Extensions:*

1**. VQ04 – Confirm Identity**

*Exceptions****:***

1. The option to reset the password is inactive.
2. The user table in the database is inactive.

*Related Use Cases:* **VQ04 – Confirm Identity.**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ04 – Confirm Identity**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-conditions:*

1. Web Page has been activated.
2. User already has an active access account.
3. User accesses the login option.
4. User accesses the forget password option.

*Description:*

1. Use case begins when the user is prompted with a template for data entry.
2. The user shall enter his username, which is the email.
3. Use case ends when user is taken to VQ03 – Reset Password.

*Relevant requirements:* none.

*Post-conditions:* The user is taken back to VQ03 – Reset Password.

*Alternative Courses of Action:*

1. In step 1 of the Description section the user can cancel the request.
2. In step 3 of the Description section the ID and/or email are not found.

*Exceptions****:***

1. The system is unavailable.

*Related Use Cases:* VQ03 Reset Password.

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

*Use Case ID:* **VQ05 - Register New User**

*Actors:* Venue Unregistered Visitor

*Pre-conditions:*

1. Web page has been activated

*Description:*

1. Use case begins when guest user accesses the register option on the Website.

2. The system shall provide the guest user or new administrator with a template for data entry.

3. The guest user should enter the following data: first Name, last name, password, security question, security answer, phone number (for confirmation).

4. The guest user should send the request by accessing the submit option.

5. The system shall then notify the guest user if the request was submitted successfully.

6. When the request is received the system shall update the record for that new user.

7. Use case ends when the record is updated.

*Relevant requirements:*

1. A guest user will only be able to register if they do not have a profile created from before.
2. Registered users (including administrators) will be rejected.

*Post-conditions:*

1. The record for that user has been created.

*Alternative Courses of Action:*

1. In step 4 o the f Description section the user has the option to cancel the request.
2. In step 5 o the f Description section, if any of the required fields are left blank the system shall notify the user of the missing fields.

*Exceptions:*

1. The register option on the webpage is not active.
2. After the user enters the required information the system the user gets rejected.

*Use Cases:* None.

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ06 – Select Ride**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-condition:*

1.      Web page has been activated.

*Description:*

1.      Use case begins when user accesses the option to select a ride.

2.      The system shall display a list of the rides available for the user.

3.      The user shall select the ride/event they want.

4.      User shall accept the selection.

5.      Use case ends when the system prompts the user with another form to finalize adding their ride (See Use case **VQ07 – Add Ride**).

*Post-conditions:*

1. The system shall update the changes made on the database.

*Alternative Courses of Action:*

1. In step 3 of the Description section the user has the option to cancel the request.
2. In Step 4 of Description section the system notifies that the request could not be completed.

*Exceptions:*

1. The option to select ride/event is not active.
2. The system could not continue after submitting the selection.

*Related Uses case:* **VQ07 – Add Ride**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ07 – Add Ride**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-condition:*

1.      Web page has been activated.

2.      Venue Registered User has an active account within the system.

*Description:*

1. Use case begins when user is presented with a template to add a ride.
2. User shall choose the time of notification from the list.
3. User shall continue by adding the ride requested.
4. The system shall notify the user time of waiting and if ride was successfully added.
5. Use case ends when customer accepts the notification displayed and user is taken to his/her activity page.

*Post-conditions:*

1. The system shall update the changes made on the database.
2. The number of rides for the customer is increased by one in the database.
3. A unique id is created on the on the queue ride table for that customer.

*Alternative Courses of Action:*

1. In step 2 of the Description section the user has the option to cancel the ride.
2. In Step 3 of Description section the system prompt the user to enter more data if a required field is left blank.
3. In Step 5 of Description section the system notifies that the request could not be completed.

*Exceptions:*

1.      The option to add ride is not active.

2.      The option to accept the notification is not active.

3.      The system could not submit the request.

*Related Uses case:* **VQ06 – Select Ride**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ08 - Edit User's Profile**

*Actors:* Single Venue Admin.

*Pre-conditions:*

1. User has an existing profile.
2. Web page has been activated.
3. User (administrator) has successfully logged onto the system.

*Description:*

1. Use case begins when *Admin search the user to be edited.*
2. Admin accesses the User Profile page and select edit option.
3. Admin is prompted with a data entry template with already loaded user info.
4. Admin can change any field they want (making sure all fields have an entry).
5. User shall accept “Update”.
6. The system shall notify the user if the request was submitted successfully.
7. Admin is taken back to the profile page.
8. Use case ends when the system saves the updated information.

*Relevant requirements:* None.

*Post-conditions:*

1. User's profile is updated in the database.

*Alternative Courses of Action:*

1. In Step 4 of Description section the user has the option to cancel the request.
2. In Step 4 of Description section the system prompt the user to enter more data if a required field is left blank.
3. In Step 5 of Description section the system notifies that the request could not be completed.

*Exceptions:*

1. The update user profile link is not active.
2. The update button is not active.

*Related Use Cases:* None

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ09 - Disable User's Account**

*Actors:* Single Venue Admin.

*Pre-conditions:*

1. User (administrator) has an existing profile.

2. Web page has been activated.

3. User has successfully logged onto the system.

4. User is on his profile page.

*Description:*

1. Use case begins when *Admin search the user to be disabled.*
2. Admin accesses the User Profile page.
3. *Admin* shall access the “Disable” option.
4. The system shall ask the *Admin* if they are sure they want to proceed.
5. The *Admin* shall confirm by accepting the “Yes” option.
6. The system shall notify the *Admin* if the request was submitted successfully.
7. *Admin* is taken back to the Main page.
8. Use case ends when the system saves the updated information.

*Relevant requirements:* none.

*Post-conditions:*

1. Record for user is updated in the database.

*Alternative Courses of Action:*

1. In step 3 of Description section the *Admin* has the option to cancel the request.

*Exceptions:*

1. The system is unavailable.
2. The “Update Profile” option is unavailable.
3. The “Disable” option is unavailable.

*Related Use Cases:* **VQ01 - User Login**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ10 - Enable User's Account**

*Actors:* Single Venue Admin.

*Pre-conditions:*

1. User (administrator) has an existing profile.

2. Web page has been activated.

3. *Admin* has successfully logged onto the system.

4. *Admin* is on his profile page.

*Description:*

1. Use case begins when admin search for the user to be enabled
2. *Admin* finds the user and accesses the "Update User Profile" option.
3. *Admin* is prompted with a data entry template.
4. *Admin* shall access the “enable” option.
5. The system shall ask the *Admin* if they are sure they want to proceed.
6. The *Admin* shall confirm by accepting the “Yes” option.
7. The system shall notify the user if the request was submitted successfully.
8. *Admin* is taken back to the Main page.
9. Use case ends when the system saves the updated information.

*Relevant requirements:* none.

*Post-conditions:*

1. Record for user is updated in the database.

*Alternative Courses of Action:*

1. In step 3 of Description section the *Admin* has the option to cancel the request.

*Exceptions:*

1. The system is unavailable.
2. The “Update Profile” option is unavailable.
3. The “Enable” option is unavailable.

*Related Use Cases:* **VQ01 - User Login**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ11 – Delete a Registered Ride**

*Actors:* VenueRegistered Visitor, Single Venue Admin.

*Pre-condition:*

1. Web page has been activated.
2. The user has successfully logged onto the system.
3. User is on his/her account page.
4. There is a user request to remove an already existing ride from his/her account.

*Description:*

1. Use case begins when user selects the option to remove an already existing ride from his/her list.
2. The system shall allow the user to select the cancel option from the specific ride selected.
3. The system shall prompt the user to confirm the option to remove the ride and shall accept.
4. Use case ends when the system displays the changes made.

*Post-conditions:*

1. The system shall update the transaction on the database.

*Alternative Courses of Action:*

1. In step 3 of the Description section the user has the option to cancel the confirmation form and make no changes to his/her account.

*Exceptions:*

1. The option to remove a ride is not active.
2. The option to confirm the changes made to the user account is not active.
3. The ride is not found on the database.

*Related Uses case:* **VQ07 – Add Ride**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ12 – View Available Rides**

*Actors:* Venue Unregistered Visitor, Venue Registered Visitor, Single Venue Admin.

*Pre-condition:*

1. Web page has been activated.

*Description:*

1. Use case begins when user accesses the option to see the rides.
2. User shall be presented with a dropdown menu.
3. Use case ends when the system displays a list of the rides available for the user.

*Alternative Courses of Action:*

1. In step 1 of the Description section the user has the option to cancel the request to see available rides.
2. In Step 2 of Description section the system does not displays the list of rides available to users.

*Exceptions:*

1. The option to see the available rides is not active.
2. The system does not display the list of rides available to users.

*Related Uses case:* None

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ13 – View User Activity**

*Actors:* Venue Registered Visitor, Single Venue Admin.

*Pre-condition:*

1. Web page has been activated.
2. The user has successfully logged onto the system.

*Description:*

1. Use case begins when user is taken to his/her account page.
2. The system shall allow the user to see his/her account page.
3. Use case ends when the system displays all info for rides added for that specific user.

*Alternative Courses of Action:*

1. In step 3 of the Description section the user has the option to close the page and not view his/her activity info.

*Exceptions:*

1. User is not taken to his/her account.
2. The system does not display user info.

*Related Uses case:* None

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*Use Case ID:* **VQ14 – Select Notification Time**

*Actors:* Venue Registered User, Venue Administrator

*Pre-condition:*

1. Web page has been activated.
2. The user has successfully logged onto the system.
3. The user has successfully selected the ride preferred.

*Description:*

1. Use case begins when user is presented with a template to add a ride.
2. User shall choose the time of notification from the list presented.
3. User shall complete choosing the rest of the template presented.
4. Use case ends when user accepts the notification displayed and user is taken to his/her activity page.

*Post-conditions:*

1. The system shall update the changes made on the database.

*Alternative Courses of Action:*

1. In step 1 of the Description section the user has the option to close the template.
2. In Step 3 of Description section the system prompt the user to enter more data if a required field is left blank.
3. In Step 4 of Description section the system notifies that the request could not be completed.

*Exceptions:*

1. The option to accept the notification is not active.
2. The system could not submit the request.

*Related Uses case:* **VQ06 – Select Ride**

**Special Requirements:**

* **Usability**: No previous training time. System is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should be sent and saved within 3 seconds.
* **Supportability**: The system should be easy to maintain, make appropriate changes, and be correctly handled by IE, Mozilla, Chrome and Safari.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## 5.3 Appendix C – Documented Class Interfaces (code) for the subsystem(s) you will implement and the constraints.

## 5.4 Appendix D – Diary of Meetings

**Virtual Meeting 1:**

**Date:** September 2, 2014

**Start Time:** 9:16PM

**Participating members:** Kely Cid, Bernard Parenteau.

**Agenda:** Programming Background discussion.

**Virtual Meeting 2:**

**Date:** September 3, 2014

**Start Time:** 12:45PM

**Participating members:** Kely Cid, Bernard Parenteau.

**Agenda:** Agreed on a optimized website development for the Virtual Queue System, and discussed Virtual Queue system requirements.

**Virtual Meeting 3:**

**Date:** September 3, 2014

**Start Time:** 3:48PM

**Participating members:** Kely Cid, Bernard Parenteau.

**Agenda:** Agreed to go forward with project requirements, technologies and tools.

**Meeting 4:**

**Date:** September 3, 2014

**Start Time:** 6:20PM

**End Time:** 8:35PM

**Participating members:** Kely Cid.

**Agenda:** Understanding of the project objectives and goals, and started working on Chapter 1 & 2 of the feasibility and project plan document.

**Meeting 5:**

**Date:** September 4, 2014

**Start Time:** 2:10PM

**End Time:** 5:30PM

**Participating members:** Kely Cid.

**Agenda:** Continue working on Chapter 2 of the feasibility and project plan document, and chapter 1 of requirement document.

**Meeting 6:**

**Date:** September 5, 2014

**Start Time:** 2:00PM

**End Time:** 6:00PM

**Participating members:** Kely Cid.

**Agenda:** Continue working on Chapter 2 & 3 of the requirement document, and chapter 3 of feasibility and project plan document. Created Trello Account.

**Meeting 7:**

**Date:** September 6, 2014

**Start Time:** 1:00PM

**End Time:** 7:10PM

**Participating members:** Kely Cid.

**Agenda:** Continue working on Chapter 3, 4, 5, and part of 6 of the requirement document.

**Meeting 8:**

**Date:** September 7, 2014

**Start Time:** 11:00AM

**End Time:** 12:45PM

**Participating members:** Kely Cid.

**Agenda:** Update Trello Account, create Power Point presentation.

**Meeting 9:**

**Date**: September 14, 2014

**Start Time**: 10:30AM

**End Time**: 7:50PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Discussed and Worked on UI design. Update mentor on project. Uploaded mockups for approval on Trello account of Login, Select Ride, and Reset Password and received feedback for mockups.

**Meeting 10:**

**Date**: September 16, 2014

**Start Time**: 10:30AM

**End Time**: 7:50PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Update mentor on project. Define views, controllers, models and url end point contracts for server side end points and test it with static data before connecting UI with server side controller. Database design and implementation.

**Meeting 11:**

**Date**: September 22, 2014

**Start Time:** 1:00PM

**End Time**: 10:45PM

**Participating members**: Kely Cid.

**Agenda**: Keep working on UI design. Added dynamic features to pages. Created packages to group all controllers, services, and DAO's. Services and DAOs interfaces and implementation for each one were created. Posted half of use cases on Trello for approval. Received approval of Use case diagram V2 for VQ system.Uploaded code to GitHub account.

**Meeting 12:**

**Date**: September 23, 2014

**Start Time:** 1:00PM

**End Time**: 10:45PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Update mentor on project. Discuss Mockups design and use cases for VQ system were uploaded to Trello.

**Meeting 13:**

**Date**: September 25, 2014

**Start Time**: 10:30AM

**End Time**: 7:50PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Update mentor on changes for Feasibility document and feasibility matrix and updates on provided VM with all tools needed already installed.

**Meeting 14:**

**Date**: September 30, 2014

**Start Time**: 10:30AM

**End Time**: 7:50PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Approval received for Mockups designs.

**Meeting 15:**

**Date**: October 2, 2014

**Start Time**: 10:00AM

**End Time**: 8:00PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Discussed Register and Login functionalities and started working on it to finish it. Started working on Design document and discuss and asked for approval of use cases. Updated Requirement document and sequence diagrams.

**Meeting 16:**

**Date**: October 7, 2014

**Start Time**: 9:30AM

**End Time**: 5:30PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Preliminary UI design approved on Trello.

**Meeting 17:**

**Date**: October 10, 2014

**Start Time**: 9:30AM

**End Time**: 5:30PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Discuss and understand the process of Selecting and Adding a Ride by a user functionalities and all validations to take into account for ride and each user.

**Meeting 18:**

**Date**: October 15, 2014

**Start Time**: 11:00AM

**End Time**: 6:25PM

**Participating members**: Kely Cid.

**Agenda**: Continue working on the Feasibility document, Requirement document, and Design document and functionalities of the system. Update mentor on project.

**Meeting 19:**

**Date**: October 20, 2014

**Start Time**: 9:30AM

**End Time**: 5:30PM

**Participating members**: Kely Cid, Bernard Parenteau.

**Agenda**: Update mentor on project. Continue working on the Feasibility document, Requirement document, and Design document to finish them. Worked on code implementation for logout and reset password and validation rules for adding a ride by a venue user. Added the rest of the use cases to Trello for approval.

# 6. References