**Feasibility Study and Project Plan**

CIS 4911 – Senior Project

Virtual Queue

**Member:**

Kenneth Kon

Michael Lazo

**Instructor**

Masoud Sadjadi

**Mentor**

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

**Date**

February 13th 2015

Copyright © Florida International University 2015. 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. The Software is provided “AS IS”, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, if no event shall the authors or copyright holders be liable for any claim, damage or other liability.

**ABSTRACT**

The Virtual Queue System will be designed for theme parks and other businesses that have multiple rides or events for which patrons typically wait in line. The idea is that both the theme park and the patron would benefit by the patrons walking around the park (and maybe spending money) rather than standing in line.

The Feasibility Study and Project Plan document gives an introduction to the Virtual Queue System. Chapter 1 gives a basic introduction about the Virtual Queue System, including the problem definition, background on the problem, definition of important terms, and an overview of the document. Chapter 2 introduces the purpose of our system since there is not an actual system, and it will also list the high-level user requirements along with an analysis of alternative solutions to the problem. Chapter 3 includes the Project Plan, which contains project management concepts of the project, hardware and software resources used, and mention the tasks, milestones, and deliverables. Chapter 4 covers different charts and information of the project, a feasibility and cost matrix, and a diary of meetings. Finally, Chapter 5 contains references to any other documents that have been used for reference.

Contents

1. Introduction 5

1.1. Problem definition 5

1.2. Background 5

1.3. Definitions, Acronyms, and Abbreviations 6

1.4. Overview of document 6

2. Feasibility Study 7

2.1. Description of Current System (Limitations and Constraints) 7

2.2. Purpose of New System 7

2.3. High-level Definition of User Requirements 8

2.4. Alternatives Solutions 9

2.4.1. Description of Alternatives 9

2.4.2. Selection Criteria 9

2.4.3. Analysis of Alternatives (See Appendix B – Feasibility Matrix) 10

2.5. Recommendations 10

3.1. Project Organization 11

3.1.1. Project Personnel Organization 11

3.1.2. Hardware and Software Resources 11

3.2 Identification of Tasks, Milestones and Deliverables (work breakdown) 13

4. Appendix 17

4.1 Appendix A - Project schedule (Gantt chart or PERT Chart) 17

4.2. Appendix B – Feasibility Matrix 24

4.3. Appendix C – Cost Matrix 25

4.4. Appendix D - Diary of Meetings 26

5. References 34

# 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. Also, the design methodology will also be given, which will include the software process models and types of models. 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. Background

The Virtual Queue (VQ) project is being proposed by Bernard Parenteau with the project proposal affiliation of Florida Logic as one of the assigned projects for the Senior Project class. It is a project that will be develop for the first time this year, in Fall semester of 2014. It will be created with features that will greatly benefit the customers as well as the theme part or other venue using the system. Furthermore, it will improve the quality time is being spend at the park or event by customers, and hopefully the sales of the theme park will increase as expected.

## 1.3. Definitions, Acronyms, and Abbreviations

**Definitions**

- **Customer**: A person or organization that buys goods or services from a store or business.

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

- **Alternative:** A possible manner by which a given problem may be resolved.

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

**Acronyms**

- **VQ**: Virtual Queue

- **FIU:** Florida International University

- **SCIS:** School of Computing & Information Sciences

- **JDBC**: Java Database Connection

- **DAO**: Database Accessing Objects

**Abbreviations**

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

## 1.4. Overview of document

The Feasibility Study and Project Plan covers several aspects of the Virtual Queue project. Chapter 1 covers general information such as problem statement, background information and definitions for this project. Chapter 2 encloses the feasibility study with an overview of the system that will be implemented, and the description and limitations of the current system. Chapter 3 specifies the project and hardware and software requirements. Chapter 4 contains the Appendix, in which miscellaneous information, such as charts and tables are shown. Finally, Chapter 5 contains works used as references.

# 2. Feasibility Study

The feasibility study chapter explores the idea of a virtual queue from a practical point of view. Initially, it considers the fact that there is no system developed in charged of completing the desired tasks. In addition, it describes the purpose of the Virtual Queue, and how the features of the VQ system will improve quality time for customers and improve business for theme parks. Following, high-level user requirements are described. Furthermore, alternatives to certain aspects of VQ system are analyzed.

## 2.1. Description of Current System (Limitations and Constraints)

The current system requires the customers to buy their tickets and wait in line at the venue until it’s their turn for their event or ride. At some theme parks, there is not even the possibility of buying the ticket for the desired ride online; it has to be purchased by the customers at the theme park. There is no mechanism to allow the user to register and sign in to see the available rides and select the rides they want; or to logout when they decide to do so; or to allow registered users to log in and logout. It lacks of a system that allows the user the ability to be added to their selected rides. It does not have a system with the ability to allow the user to see all the queues he/she registered. There is no mechanism for adding/removing registered users from the queue according to the queuing algorithm, or to allow users to remove themselves from a registered ride. There is no system with the capability of notifying the users before their selected rides/events starts.

## 2.2. Purpose of New System

Considering that time spent at a theme park or other venue is sometimes limited due to the fact that customers have to wait in line for rides; the implementation of the VQ system is a good idea.

This new system will provide the customers the ability to see which rides are available and sign in to different ones, as long as there is no time conflict. It will also give the customer the possibility to see all the rides he signed on for and received a notification when their turn is coming on, so they can go to their selected ride. Therefore, providing this system, customers will have a more enjoyable experience at the venue by not having to wait in line, and the venue will also benefit by customers shopping or enjoying other amenities rather than waiting in line.

Overall, the main purpose of the new system is to add functionalities and new features to the current system, which will greatly benefit both the customers and the park/event place.

## 2.3. High-level Definition of User Requirements

The user requirements for using the VQ system are very minimal. The user would be required to have a smartphone, tablet or any other device with web access in order to use the web application. In addition, the new system shall…

Allow unregistered users to register.

Allow users to validate their account.

Allow registered users to delete their accounts.

Allow registered users to sign in to their accounts.

Allow registered users to logout of their account.

Allow users to reset their password.

Allow registered users to have access to the available rides/events.

Allow registered users to be added to their selected rides/events.

Allow registered users to see all the rides/events they signed on for.

Allow registered users to delete themselves from registered rides/events.

Allow the user to be notified before their rides/events time occurs.

Allow the system to add and/or remove registered users from their selected rides.

Allow the system to send notification to registered user before ride time added approaches.

Allow the system to store and retrieve information regarding to the rides/events.

Allow the system to check for duplicates registration or multiple registrations for same user.

Allow the system to handle network connectivity issues.

## 2.4. Alternatives Solutions

The sections 2.4.1 throughout 2.4.3 describe the alternative implementations for this VQ system project.

### 2.4.1. Description of Alternatives

Alternative 1: System will be built from scratch. This will give the ability and control required over the architecture of the system. Also, all features will be implemented from the beginning. There will be no need of modifying any existing code, which will increase the efficiency of programming. In addition, it will have a vast set of features, which means that more time is going to be needed to develop and implement all features. Although, the system will perform a complete set of operations, and the features are going to be designed as there were initially designed to be.

Alternative 2: A new system will have to be built from scratch. This will give the ability and control required over the architecture of the system. The system will be implemented with a basic set of features for the first release, and the features are going to added as the system is updated throughout each version. Also, all features will be implemented from the beginning. There will be no need of modifying any existing code, which will increase the efficiency of programming.

### 2.4.2. Selection Criteria

The selected criteria that is going to be used for the consideration of the environment software suite alternatives are mentioned and explained below:

● **Operational Feasibility**: Ability of the users, developers and those involved with the project to use and support the proposed system.

● **Technical Feasibility**: Reliability of the software and hardware and its capability to provide the intended functionality of the system

● **Schedule Feasibility**: How long the system would take to be designed and implemented

● **Economic Feasibility**: Ability of the system to cover its development and maintenance costs after its completion

### 2.4.3. Analysis of Alternatives (See Appendix B – Feasibility Matrix)

The alternatives described above are the options available for this project. Taking into account this is the first version of this system, there is no other platform that was provided to modify or extend. Therefore, starting this project implementation from the beginning is the only and best option to develop the system and implement all new basic functionalities. In addition, there is no enough time throughout the whole semester, and just one person was assigned to work on this project. Refer to Appendix B for the Feasibility Matrix.

## 2.5. Recommendations

Taking into account the solutions provided, the conclusion is to start building the project from the beginning. Since there is no previous platforms implemented that can be take into consideration, the only and best solution is to start from scratch. Also, there is not enough time, and there is just one person assign to the project to implement all functionalities and have all features perfectly working; so the best idea is to provide a system that will have just the basic functionalities. New features upgrades can be done and added throughout the lifespan of the project.

**3. Project Plan**

This chapter introduces the VQ system a project management perspective. First of all, the project organization that is going to be used will be described, with all roles listed. Following, hardware and software required in order to develop the project will be listed. Finally, tasks, milestones, and deliverables will also be listed.

## 3.1. Project Organization

For this project, I will be in charge of all functionality and roles of the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Roles | Tasks | Periods required | Key Phases |
| Kenneth Kon | • Developer • Scrum master (ALT.) • Test Engineer • Document Editor | • Project Status  • Schedule/Task • Implementation • Testing • Project Documentation | 1/12/15 to 05/01/15 | All |
| Michael Lazo | • Developer • Scrum master (ALT.) • Test Engineer • Document Editor | • Project Status  • Schedule/Task • Implementation • Testing • Project Documentation | 1/12/15 to 05/01/15 | All |

### 3.1.1. Project Personnel Organization

All sections of this project will be assigned to the only member of the group, which will be in charge to develop all the components needed to support the different functionalities.

### 3.1.2. Hardware and Software Resources

**Hardware**

The following are the specification of the computer that will be used for the development of this project:

**Laptop with:**

**- Processing Power:** 1.8 GHz Intel Core i5

**- RAM Memory:** 4 GB 1600 MHz DDR3

**- Available space on hard drive:** 128 GB flash storage

**Other Devices**

- **Input devices**

a) Keyboard- Standard keyboard

b) Mouse- Trackpad mouse

- **Output devices**

a) Monitor display- 13-inch (1440 x 990) Intel HD Graphics 4000 graphics

**Software**

1. **Safari 7.0.6:** last, most updated version of the Safari browser that will be used to test Virtual Queue.
2. **Google Chrome 37.0:** last, most updated version of the Google Chrome browser that will be used to test Virtual Queue.
3. **Spring Framework 4.0:** last, most updated version of Spring Framework, an MVC-based, Java framework used for development. It will be used to develop the front-end and back-end of Virtual Queue.
4. **Quartz Scheduler**: open source job scheduling library that can be integrated within virtually any Java application. It will be used to create, start, stop jobs, and send notifications and delete users from rides.
5. **Bootstrap 3**: most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. It will be used for the front-end design development.
6. **Spring JDBC template**: make database call access to MySQL database.
7. **JQuery**: Fast, small, and feature-rich JavaScript library. It will be used to connect the front-end development with the back-end development.
8. **MySQL**: database that will be used to store data for the Virtual Queue system.
9. **StarUML:** UML software platform that will be used to create diagrams for the document.
10. **VMWare:** virtual machine software that will be used for the deployment of the software.
11. **Eclipse Luna:** IDETool software used to write the java classes for the Virtual Queue system.
12. **JBoss 7.1.1**: Java J2EE Application Server used to deploy and run the Virtual Queue system.

## 3.2 Identification of Tasks, Milestones and Deliverables (work breakdown)

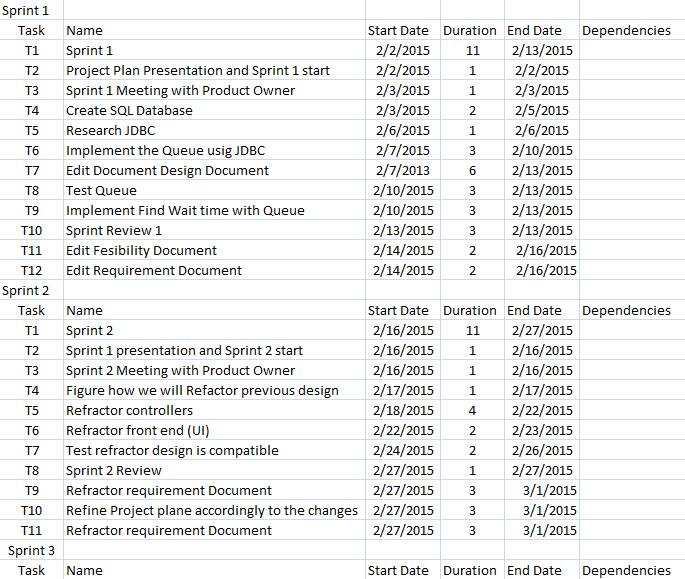
Below is a list of all different tasks, milestones, and deliverables for the project, which can be changed according to the development of the project:

|  |  |
| --- | --- |
| Milestones | Task & Deliverables |
| Documentation | * Feasibility Study * Requirement Document * Design Document * Deliverable 4 |
| Environment Setup | * IDE Installation (Eclipse) * JDK 8 Install * StartUML Install * Source Tree for Github * Jboss Installation and Configuration * Maven Project Setup * Spring framework setup * Project dependencies and third parties libraries * Dependency injections setup * MySQL install |
| * Development Tasks: | |
| UI Design | * Create main page template view * Create login page template view * Create My Account page template view * Create Add ride to user page template view * Create Select User template view * Create reset password page template view * Create Add User page template view |
| Database Design and Implementation | * Create Tables * Create tables relationships and constrains * Create EER diagram |
| Main Page Functionality | * Add dynamic content to main page. (JQuery, Ajax calls) * Add server side implementation to main page. * Create main controller with URL definitions and contracts * Create main service interface/implementation * Create main DAO for data access. |
| Login and Logout functionalities | * Add dynamic content to login page (JQuery, Ajax calls) * Server Side Implementation. * Create login controller with URL definitions and UI contracts * Login controller unit test. * Create user Bean. * Unit test to user bean. * Add user information to session when user log * Create login service interface/Implementation * Login service unit test * Create login DAO to facilitate data access * DAO unit test. * Add logout controller with URL contract * Logout controller unit test. * Remove user information from session data. * Unit test. |
| Add User Functionality | * Add dynamic content to user page (JQuery, Ajax calls) * Server Side Implementation. * Create User controller with URL definitions and UI contracts * User controller unit test. * Add User information to session when user log * Create User service interface/Implementation * User service unit test * Create User DAO to facilitate data access * DAO unit test. |
| Reset Password Functionality | * Add dynamic content to user page to reset password (JQuery, Ajax calls) * Server Side Implementation. * Add method User controller with URL definitions and UI contracts for password reset. * User controller unit test. * Add functionality to User service interface/Implementation * User service unit test * Add functionality to User DAO to facilitate data access * DAO unit test. |
| User Ride Functionality | * Add dynamic content to user Ride page (JQuery, Ajax calls). * Add UI data validations. * Server Side Implementation. * Create login controller with URL definitions and UI contracts * Add controller unit test * Create Ride Bean. * Unit test to ride bean. * Add user information to session when user log * Create user ride interface/Implementation * Unit test for service interfaces * Create User Ride DAO to facilitate data access * DAO unit test. * Add User Validator interface when registering for a ride. * Validator unit test. * Add User validator implementation with rule definitions. * Validator implementation unit test for each method. * =================================== * Add Ride Registration Business Rules Interface * Add Ride Registration Business Rule implementation. * Add Unit test for Ride Registration * Add Rules Interfaces * Add Rules Implementations. * Add unit test to Rules implementations. * Add Rule Builder Class. * Unit test to Rule Builder. |
| Admin functionalities | * Create UI design for Login and Search modal for admin user. * Server Side Implementation. * Add method Admin controller with URL definitions and UI contracts for login. * Add functionality to Admin service interface/Implementation * Add functionality to Admin DAO to facilitate data access. |

# 4. Appendix

This chapter contains a Gantt Chart with the scheduled time of work for the whole project, a feasibility matrix, a cost matrix that shows the estimate total cost for the project, and a diary of meetings.

## 4.1 Appendix A - Project schedule (Gantt chart or PERT Chart)



**Sprint 1 Gannt Chart**

**Sprint 2 Gantt Chart**

## 4.2. Appendix B – Feasibility Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Feasibility Criteria** | **Weight.** | **Alternative 1** | **Alternative 2** |
| **Operational Feasibility**: Ability of the users, developers and those involved with the project to use and support the proposed system. | **30%** | Fully supports required functionalities.  **Score: 100** | Fully supports required functionalities.  **Score: 100** |
| **Technical Feasibility**:  **Technology: An assessment of the maturity, availability, ability to acquire, and desirability of computer technology needed to support this candidate.**  **Expertise: An assessment of the technical expertise needed to develop, operate, and maintain the candidate system.** | **30%** | All platforms would be supported including smartphones platforms.    Bootstrap 3 and JQuery will be the primary language and Eclipse Luna will be extensively used. There is a reasanoble amount of experience using HTML, CSS, MySql, which will be the main technologies in use.  **Score: 85** | All platforms would be supported including smartphones platforms.    Bootstrap 3 and JQuery will be the primary language and Eclipse Luna will be extensively used. There is a reasanoble amount of experience using HTML, CSS, MySql, which will be the main technologies in use.    **Score: 85** |
| **Economic Feasibility**: Ability of the system to cover its development and maintenance costs after its completion. | **30%** | The new platform will be built using open resources, self database hosting and school and developer’s resources, which will make the system cost $0. There will no be payback  **Score: 100** | The new platform will be built using open resources, self database hosting and school and developer’s resources, which will make the system cost $0. There will no be payback  **Score: 100** |
| **Schedule Feasibility**: How long the system would take to be designed and implemented | **10%** | Based on the fact that the semester is short and just one person is assigned to it, the design and implementation of all set of operations of the system as they were initially design to be is not feasible in the timeframe.    **Score: 30** | Based on the experience of the team and questions asked to the client within the project, the design and implementation of the project with the basis functionalities is quite feasible in the timeframe.  **Score: 90** |
| **Ranking:** | **100%** | **78.75** | **93.75** |

## 4.3. Appendix C – Cost Matrix

|  |  |  |
| --- | --- | --- |
| Resources | Quantity | Cost |
| PC (Hardware) | 1 | $0.00 |
| MySQL | 1 | $0.00 |
| Spring Framework |  | $0.00 |
| JQuery Library |  | $0.00 |
| Bootstrap 3 Framework |  | $0.00 |
| Quartz Scheduler |  | $0.00 |
| Development |  | $0.00 |
| Testing |  | $0.00 |
| Total Costs: |  | $0.00 |

## 4.4. Appendix D - Diary of Meetings

**Meeting 1**

6:15PM-7:15PM

1/22/2015

Michael Lazo - scrum master

Kenneth Kon

Bernard Parenteau

Topic:

Talking about the requirements with Bernard for Virtual Queue 2.0

**Meeting 2**

6:15PM-7:15PM

1/29/2015

Michael Lazo

Kenneth Kon - scrum master

Bernard Parenteau

Topic:

Discussed about the Database layout and the fields of Virtual Queue 2.0

**Meeting 3**

6:15PM-7:15PM

2/03/2015

Michael Lazo - scrum master

Kenneth Kon

Bernard Parenteau

Topic:

Discussed on what User Story we should work on with the Product Owner/Mentor, for Sprint 1.

**Meeting 4**

6:15PM-7:15PM

2/09/2015

Michael Lazo

Kenneth Kon - scrum master

Bernard Parenteau

Topic:

Discussed our progress on the Sprint 1, discussed any impediments.

**Meeting 5**

6:15PM-7:15PM

2/13/2015

Michael Lazo

Kenneth Kon - scrum master

Bernard Parenteau

Topic:

Discussed Sprint 1 Review, if User Story was satisfy the requirements.

Discussed also discussed the impediments and need to refactor the previous design.

**Meeting 6**

6:15PM-7:15PM

2/17/2015

Michael Lazo

Kenneth Kon

Bernard Parenteau - scrum master

Topic:

Discussed Sprint 2 Planning, convince Product owner Team Story has higher priority.

**Meeting 7**

6:15PM-7:15PM

2/28/2015

Michael Lazo

Kenneth Kon - scrum master

Bernard Parenteau

Topic:

Discussed Sprint 2 Review, display the new design of the Refactored product.

Discussed also discussed the impediments and need to Refactor the previous design.

# 5. References

Images used:

1. <http://www.plan-family-reunions.com/themeParks.html>
2. http://ru.forwallpaper.com/wallpaper/abstract-circles-patterns-dots-light-colors-bokeh-abstraction-172728.html

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Roles | Tasks | Periods required | Key Phases |
| Kenneth Kon | • Developer • Scrum master (ALT.) • Test Engineer • Document Editor | • Project Status  • Schedule/Task • Implementation • Testing • Project Documentation | 1/12/15 to 05/01/15 | All |
| Michael Lazo | • Developer • Scrum master (ALT.) • Test Engineer • Document Editor | • Project Status  • Schedule/Task • Implementation • Testing • Project Documentation | 1/12/15 to 05/01/15 | All |

|  |  |
| --- | --- |
| Artifacts | Priority (1-10) |
| Queue Implementation | 10 |
| Database Management | 10 |
| Send Notifications | 6 |
| Administrative Rights | 3 |
| Build Ride | 7 |
| Simulator | 9 |
| Find Wait Times | 7 |
| Dequeue | 10 |
| Status Request | 4 |
| Test Queue Controller | 6 |