Software Requirements Specification

for

Fernwood Farmers' Market
Booth Scheduling System

or

FaBS

GROUP 11

2015/05/19

Change History

Name	Date	Reason for Changes	Version	Updated By
Initial Draft	**2015/05/19**	**n/a**	**0.1**	**Group 11**
Draft 0.2	**2015/05/21**	**paragraph development**	**0.2**	**Group 11**
Draft 0.3	**2015/05/24**	**finalizing for assignment 1 hand-in**	**0.3**	**Group 11**

Document Approvals

Name	Role	Signature

Table of Contents

1 - Introduction

1.1 Purpose

The Fernwood Farmers' Market is looking to improve their quality of service by providing a more convenient way for vendors to book booths. This document specifies the initial requirements gathered for the creation of a web-application designed to facilitate vendor to book market booths online and allow for administrators to better organize market sessions.

1.2 Document Conventions

Entity Relationship diagrams within this document uses Crow's notation.

1.3 Project Scope

The primary goal of FaBS is to allow vendors who are interested in Fernwood Farmer's Market to have greater accessibility when booking booths and consequently help minimize organizational overhead for administrators. By means of the application, vendors will be able to view current market schedules, book booths, and also cancel their bookings. Administrators would then be able to freely modify bookings, i.e. create and cancel, as well as ban and modify user accounts as required.

1.4 References

1. SRS Template: http://www.jaysonjc.com/programming/how-to-write-a-software-requirementsspecification-srs-document.html

1.5 Overview

The introduction is followed by some background information relating to the system, a detailed discussion about the use cases, complete set of features and architecture of the system. The document then concludes with a section outlining the non-functional requirements that includes performance, scalability, security, maintainability, usability, multilingualism, auditing, and availability.

2 - System Description

2.1 Product Perspective

The Fernwood Farmer's Market requires a web-application that allows vendors to book booths online, as well as administrators to manage market sessions. The application must support specific details and information about booths and vendors, a system that automates temporary blocks on vendors for late cancellations of bookings, and the ability for administrators to edit user accounts and bookings. Currently, no particular web-application suits these specific needs, hence there is a clear need for an application to be developed.

2.2 Product Features

The Farmers' Market would like to provide online bookings for the following booth types: 3 Lunch Booths, 4 Produce Booths, 5 Merchandise Booths. FaBS will allow users to book these booths and cancel bookings. Users will also be able to view a history of the booths they have used previously.

2.3 User Classes and Characteristics

FaBS currently consists of the following primary classes of users:

Vendors:

Vendors are businesses that want to book booths in the farmer's market. Vendors will have the least amount of permission of all types of users. They can only book or cancel booths, view market schedule and add profile information.

Administrators:

Capable of editing booking and account information without any restrictions. Administrators will have the ability to edit all information including specifics of individual vendor accounts.

System Operators:

Capable of overriding all rules and has the power to create new administrators. System Operators can also shut down the system and allow for development if problems arise.

2.4 Design and Implementation Constraints

The libraries available for development will be constrained to packages that are compatible with the specified operating environments.

2.5 Assumptions and dependencies

User's must have scripting enabled in their updated browser to properly view the website and use the available features.

2.6 System Technology Stack

The web application will use a Model View Controller architecture for implementation purposes. To implement this architecture, the use of databases will be strictly restricted to MongoDB; Express.js, Node.js, and JQuery will be used for backend and frontend development.

3 - Functional Requirements

3.1 System Features

3.1.1 User Booking and Cancelling

Users are able to book booths through the web application. Users are also able to cancel any bookings, but will be blocked from making another booking if they cancel a booking less than 24 hours before the booked start time.

3.1.2 User Registration and Profile Editing

Users can create a profile of their booth showing off what merchandise they display. This can be edited by the user and by administrators.

3.1.3 Booking History

Users can view the history of their events. This includes which booths they have booked and canceled in the past.

3.1.4 Administrative Control

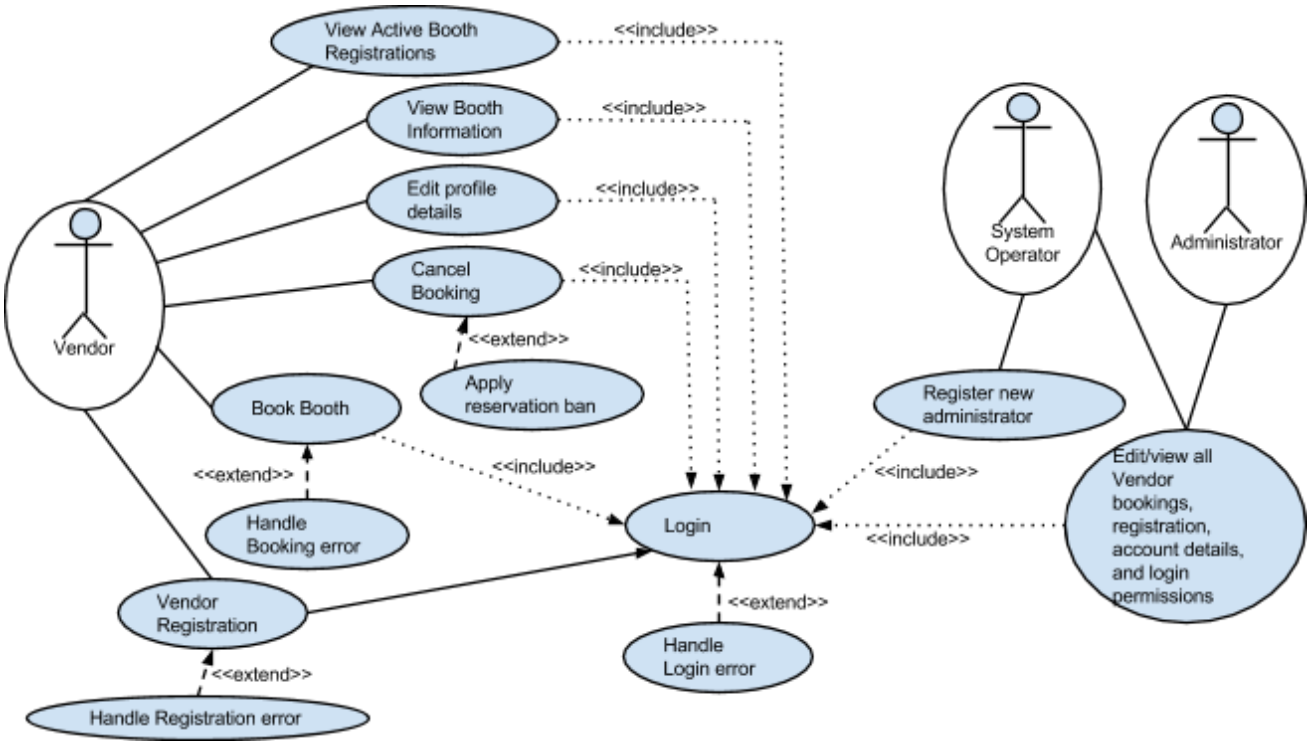
The administrator(s) can view and edit vendor profile and booking information.

3.1.5 Registration Blocks

If a vendor cancels a booked booth within 24 hours of the booking time, the vendor will be penalized with the inability of booking another booth for the next 48 hours.

3.2 Use Cases

3.2.1 Use Case Diagram



3.2.2 Use Case 1

ID	1
Description	User finds a restaurant with a dinner deal closeby
Actors	User
Preconditions	User allows the application to have access to the device's current location
Basic Steps	● User enters username/password ● User adds detail to profile
Exceptions	Username already taken. Password does not meet security requirements.
Business validations/Rules	Username should match company/product provider
Postconditions	Username defined is no longer available to new vendors

3.2.3 Use Case 2

ID	2
Description	A user books a booth for the market
Actors	Vendor or administrator or system operator
Preconditions	User must be logged in and not have any restrictions on their account that prevent them from booking a booth.
Basic Steps	User will view the market schedule, pick an open slot which matches the type of booth they plan to have, and confirm their booking.
Exceptions	The time slot has been booked while they were in the process of booking it themselves. User attempts to book a second booth.
Business validations/Rules	● Users are unable to book multiple booths ● Users are restricted to book from 10am-2pm or 4pm-8pm
Postconditions	The user is booked for a specific time slot at a specific booth.

3.2.4 Use Case 3

ID	3
Description	User edits profile details
Actors	Vendor or administrator or system operator
Preconditions	User must be logged in
Basic Steps	User will log in, then access and edit their profile information
Exceptions	Invalid information entry eg. contact email without “@”
Business validations/Rules	Profile details are not allowed to contain profanity
Postconditions	Updated email no longer available to other vendors

3.2.5 Use Case 4

ID	4
Description	User cancels a booked time slot.
Actors	Vendor or administrator or system operator
Preconditions	User must be logged in.
Basic Steps	User will view the schedule page and their scheduled bookings at the bottom, click the cancel button beside their booking, and confirm through the dialog which pops up.
Exceptions	An administrator has already canceled their booking while they were in the process of doing so. (Appointment not found).
Business validations/Rules	If vendor cancels a booking within 24 hours of the booking, the user’s account will be flagged and unable to book another booth for the next 48 hours. Rule does not apply to administrators or system operators
Postconditions	If user is a vendor and cancels 24 hours before their planned booking, their account is flagged and unable to book a booth for the next 48 hours.

3.2.6 Use Case 5

ID	5
Description	User Views others' booth bookings
Actors	Vendor or administrator or system operator
Preconditions	User must be logged in
Basic Steps	User will view the map of the market after login, showing bookings of the upcoming market
Exceptions	User unable to login
Business validations/Rules	User cannot edit/book over others' booth bookings
Postconditions	N/A

3.2.7 Use Case 6

ID	6
Description	User views a booth's information
Actors	Vendor or administrator or system operator
Preconditions	User must be logged in
Basic Steps	User will login, click on a booth, and view the information of that booth
Exceptions	User can not login
Business validations/Rules	User not able to edit other booths' information
Postconditions	N/A

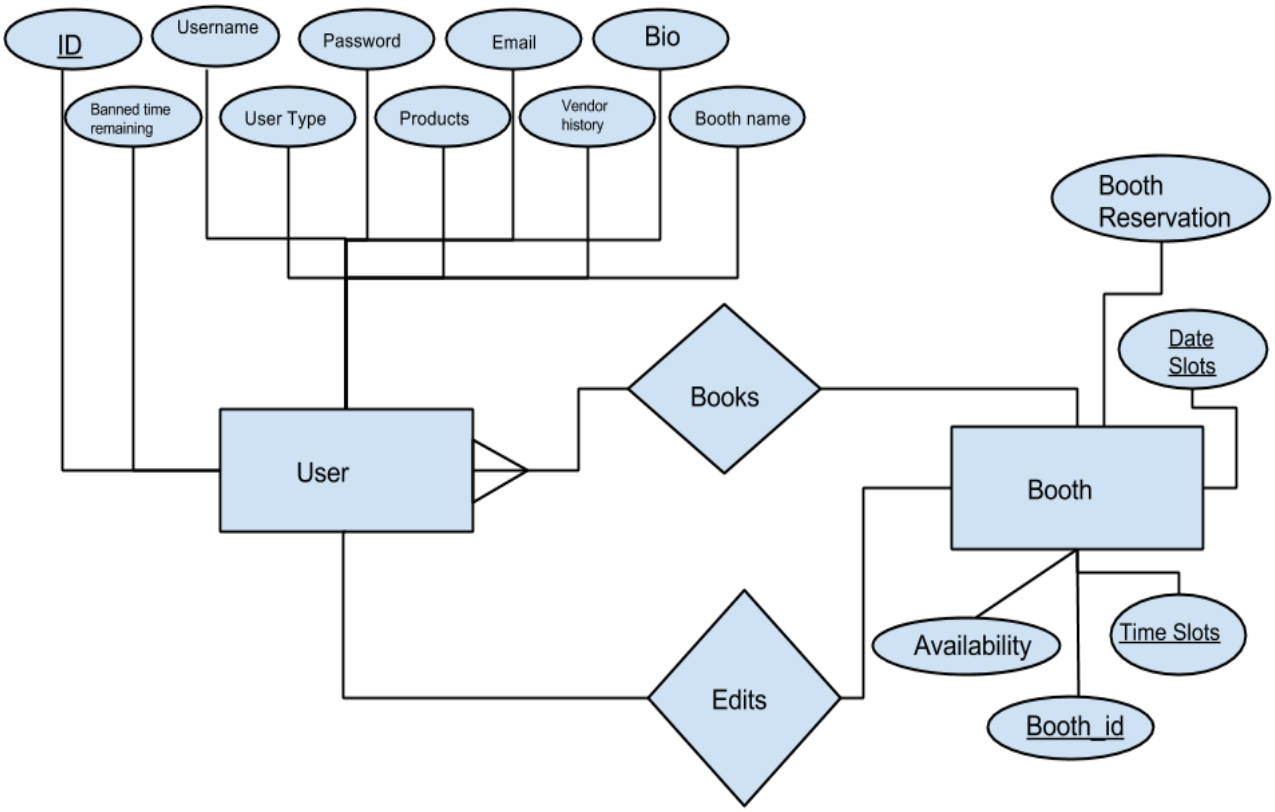
3.2.8 Use Case 7

ID	7
Description	System Operator creates a new administrator account
Actors	System Operator
Preconditions	System Operator must be logged in
Basic Steps	System Operator will click to register a new administrator, enter the administrator's username/password and permissions
Exceptions	User must have a System Operator account
Business validations/Rules	System Operator cannot make more System Operators
Postconditions	New administrator added to database

3.2.9 Use Case 8

ID	8
Description	Login to the system
Actors	Vendors, Administrators, System Operators
Preconditions	N/A
Basic Steps	User will enter their username and password in the designated boxes, and press login
Exceptions	User types in the wrong username/password combination
Business validations/Rules	User can only login to their own account
Postconditions	User will be logged in

3.3 Entity Relationship Diagram



3.4 Data Dictionary

3.4.1 User

- This is the object for Vendors, this will include their login info, and anything else associated with their account.

Attribute	Type	Optional?	Notes
Username	String	No	Username must be unique across all system users.
Password	String	No	Must be of length 8 and must contain: ● 1 uppercase character (minimum) ● 1 lowercase character (minimum) ● 1 number (minimum)
Email Address	String	No	Email address for sending confirmations
Biographic information	String	No	A short text block where the vendor can talk about their business and wares.
Booth Name	String	No	Name of the vendor's booth.
Banned time remaining	int	No	A value of 0 signifies no ban status on their account. Otherwise, the value will be the time remaining on their ban.
Booth Type	Int	No	Categorizes their booth as merchandise, produce, and/or lunch.
Products	String[x]	No	None/null if a product type is not applicable, but otherwise it will be lunch items, produce items, and merchandise items.
User Type	Int/Byte	No	Categorizes a user as a vendor, administrator, or system operator.
Vendor_id	Int	No	When booth booked, id number of vendor using booth. Unique vendor identifier.
Vendor_history	int list	No	Takes deleted booth bookings when they have passed.

3.4.2 Booth

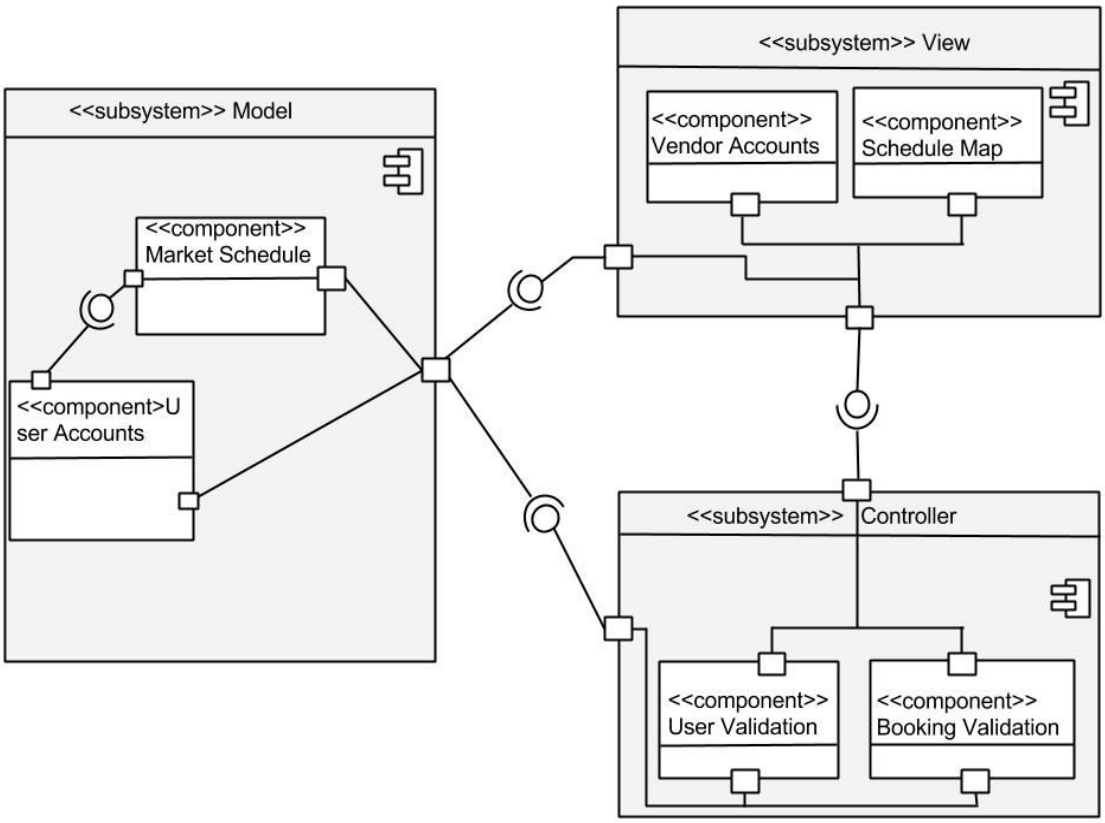
Attribute	Type	Optional ?	Notes
Availability	Boolean	No	Whether or not the booth is available at a certain time.
Booth_id	int	No	Each booth has a unique ID, eg: produce booth 2 could be 5
Time Slots	int	No	eg: 10001400 , or 16002000 (24 hour start time)(24 hour end time) this gives flexibility if they want to change schedule times
Date Slots	int	No	eg: 20150521 , or 20150130 yyyy/mm/dd
Booth Booking	int List	No	Initially empty eg: 1430-20150521-14-000562 time-date-booth-vendor unique identifier for a bookings booking

4 - System Architecture

4.1 Conceptual Architecture

The system is organized as an MVC architecture. The model component will contain the database information for accounts and the market schedule. In the view component there will be the views for viewing the market schedule and making bookings as well as the view for vendor accounts. Bookings and account changes made by users will be verified by the controller component's validation components and valid changes pushed to the relevant database in the model.

4.2 Component Diagram



5 - Technical Requirements (Non-functional)

5.1 Performance

The web-application is expected to load and transition between pages as per user requests with seemingly insignificant delay times. Calls and responses to and from databases should comply accordingly; caching and or prefetching data could be used to portray faster network speeds and allow for better performance.

5.2 Scalability

The system should scale linearly with the number of registered users and active users. The implemented architecture framework should also be open to allow for future systems integration.

5.3 Security

All information pertaining to a particular user should be confidential and only disclosed to any third party identity as per user requests. User registration and login procedures should be carried out under a secure connection and ideally performed over HTTPS. All users will have a unique user identification and all passwords will be cryptographically hashed before being stored within a database.

5.4 Maintainability

The system could use user-submitted reports on application functionality in order to maintain a satisfied user-base. Code fragments could also undertake frequent code-reviews from multiple developers to maintain industry standard software engineering practices.

5.5 Usability

The system is expected to be responsive and should support most popular web browsers including Google Chrome v42+, Firefox v38, Internet Explorer 10+ and Safari v8 whether on static or mobile devices. The implemented user-interface should be intuitive for users with little or no experience with technology and should allow for them to comfortably navigate through the application according to their needs.

5.6 Multilingual Support

The system should initially support English and sequentially provide support for French, Mandarin, and Spanish in further iterations. However, the system should also support user requests for additional languages as well.

5.7 Auditing and Logging

All system errors will be logged for later analysis to improve internal fault tolerance. To aid in security, all vendor and administrator login attempts will be recorded and logged, in addition to the IP address from which the login attempt was made. A better understanding of the location of the user base could then be estimated by logging all connected IP addresses. This could help identify and resolve various usability problems, consequently resulting in an overall improvement in user workflow.

5.8 Availability

All system services will have no planned downtime and any changes/deployments that need to be made can be done without interrupting user access. A secondary proxy server will be available in order to access the services in case of any malicious attacks on the system.

6 - Open Issues

The system will need a way of limiting how far bookings can be made in advance, preferably with a set time during the day when the bookable schedule is expanded.

There are many features that will not be implemented in the current implementation, but are planned to be implemented in future iterations. It is planned to have an interface for users who are not vendors, administrators or system administrators. This interface will display information about what vendors are available on what days, vendor biographies, and general information about the market. This interface is to promote the farmer’s market, informing consumers on what to expect from visiting the market.