

**Architecture Specification Document**  
for  
**Fernwood Farmers' Market**  
**Booth Scheduling System**  
or  
**FaBS**

# Table of Contents

## [1.0 Class Diagram](#)

## [2.0 Proposed Use of Design Patterns:](#)

### [2.1 Singleton:](#)

### [2.2 Observer:](#)

### [2.3 Facade:](#)

## [3.0 Storyboards](#)

### [Use Case 1: Register for a new account](#)

### [Use Case 2: Booking a booth](#)

### [Use Case 3: User edits profile details](#)

### [Use Case 4: Cancelling a booking](#)

#### [Use Case 4.1: Vendor Booking Cancellation:](#)

#### [Use Case 4.2 Administrator/System Operator Booking Cancellation:](#)

### [Use Case 5: User Viewing a Market Schedule](#)

### [Use Case 6: User Views Other Booth's Information](#)

### [Use Case 7: System Operator Creates New Administrator](#)

### [Use Case 8: Login to the System](#)

### [Use Case 9: Create Support Request](#)

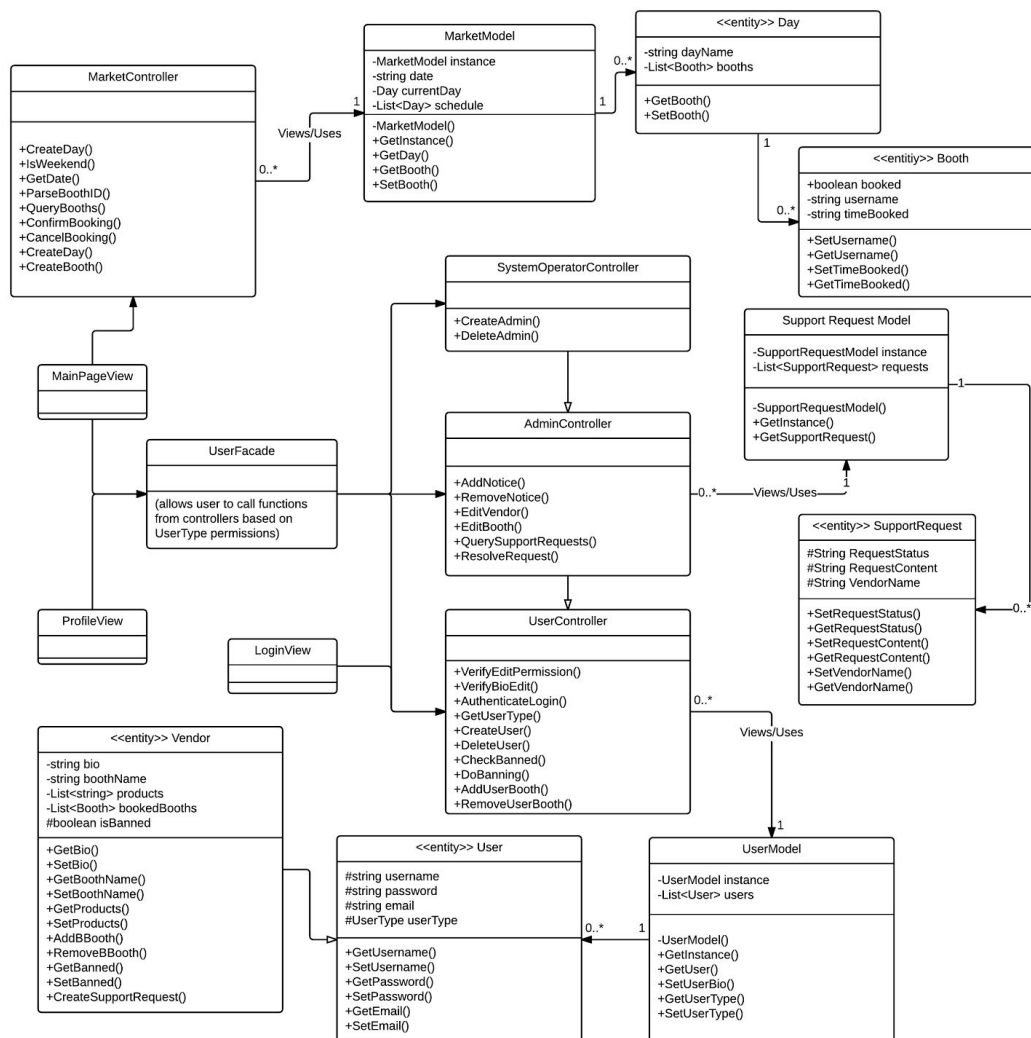
## [4.0 Sequence Diagrams](#)

### [4.1 Booking A Booth](#)

### [4.2 Cancelling A Booking](#)

### [4.3 User Edits Profile Details](#)

## 1.0 Class Diagram



**Note:** All view classes contain functionality to both render the required UI screens to the users and interact with the associated controllers. They have been excluded from the class diagram for simplicity's sake, in order to avoid confusion and ambiguity.

The system uses an MVC architecture and all major classes can be broken down into model, view, and controller components. In order to maintain transparency, changes in each major view and model takes place through their associated controllers. The MarketController in conjunction with the MainPageView takes care of all app functionality relating to viewing/booking/cancelling booths from the user's perspective; whereas, the UserController allows users to verify/edit profile information. The UserFacade allows administrators a means to comfortably carry out regulatory tasks as well as edit user information as required.

## 2.0 Proposed Use of Design Patterns:

The Fernwood Farmers' Market Booth Scheduling System (FaBS) intends to use the Singleton, Facade and Observer patterns in our implementation.

### 2.1 Singleton:

We have three singletons which serve as the model part of our MVC architecture: the `MarketModel`, `SupportRequestModel` and the `UserModel`. The `MarketModel` has multiple `Day` objects which in turn has multiple `Booth` objects. The `UserModel` consists of `Vendor` objects and objects of its inherited class, `User`. And lastly, `SupportRequestModel`, which contains a list of `SupportRequest` objects. These model classes essentially represent all stored data and as such we only want a single instance of each class working with their respectively stored data. That is to say, we only want one set of market data, one set of user data and one set of support request data.

### 2.2 Observer:

We would like to have status changes in the market schedule be updated to a user's view in real-time without them having to reload the page. The Observer design pattern can be successfully applied here so that updates in the market schedule would notify all connected users' views to update their display. Depending on whether some implementation of this is available in the MEAN stack, we may have to add a `ConnectedUsers` model which stores a list of connected users. When a change is made in one of the other models, they would then notify the `ConnectedUsers` class which would then call an `Update()` function in each connected user's view. In this case, `ConnectedUsers` would be the subject with a public `NotifyUsers()` function, while the various view classes are the observers. `ConnectedUsers` would have `AddListener()` and `RemoveListener()` functions in order to update its list of connected users.

Note: the `ConnectedUsers` class is currently not included in the class diagram.

## 2.3 Facade:

The `UserFacade` class acts as a simplified code interface for the views to interact with the `UserController`, `AdminController`, and `SystemOperatorController` classes. Depending on the logged-in user's permissions, it will delegate their actions to the appropriate class as well as dictating whether they see special Admin and System Operator UI elements.

## 3.0 Storyboards

### Use Case 1: Register for a new account

The screenshot shows a web browser window with the address bar displaying "http://fabs.org". The page content includes a logo placeholder (a rectangle with an 'X' and the word "LOGO" twice), followed by input fields for "username:" and "password:". Below these are two buttons: "Register" and "Login". A callout box with the text "User will click to register" has an arrow pointing to the "Register" button.

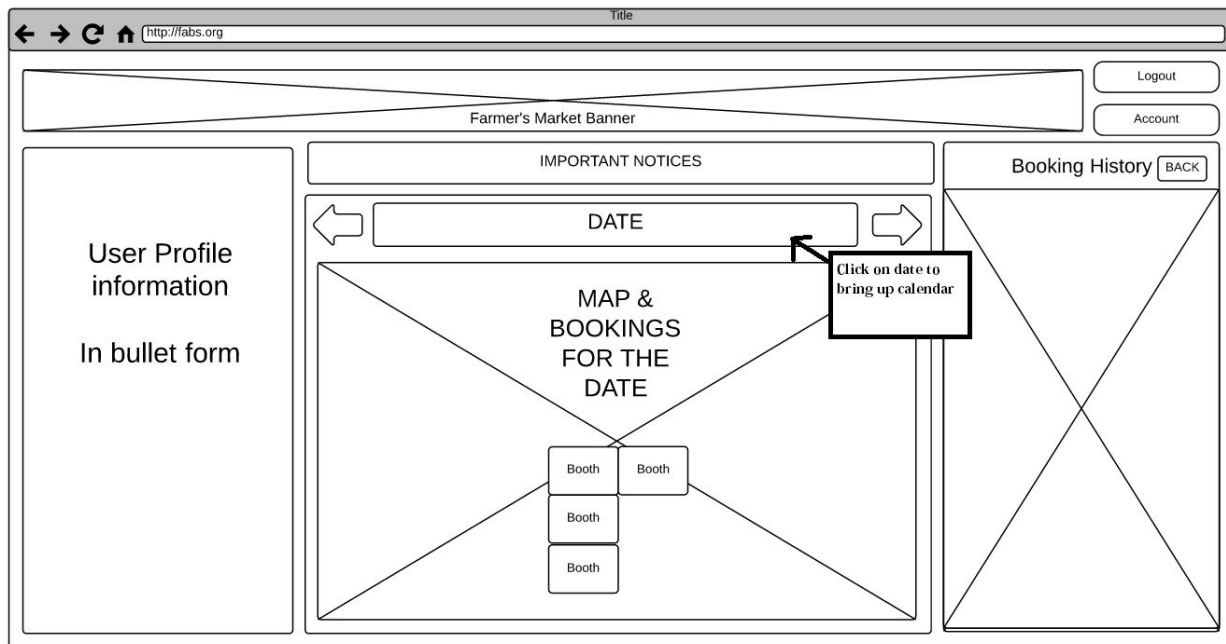
User will be shown this screen upon visiting the site. They are to click the register button to advance to a page where they can enter user information.

Here, new users must fill out all fields in order to progress. Their booth and login name must not already be taken by another user.

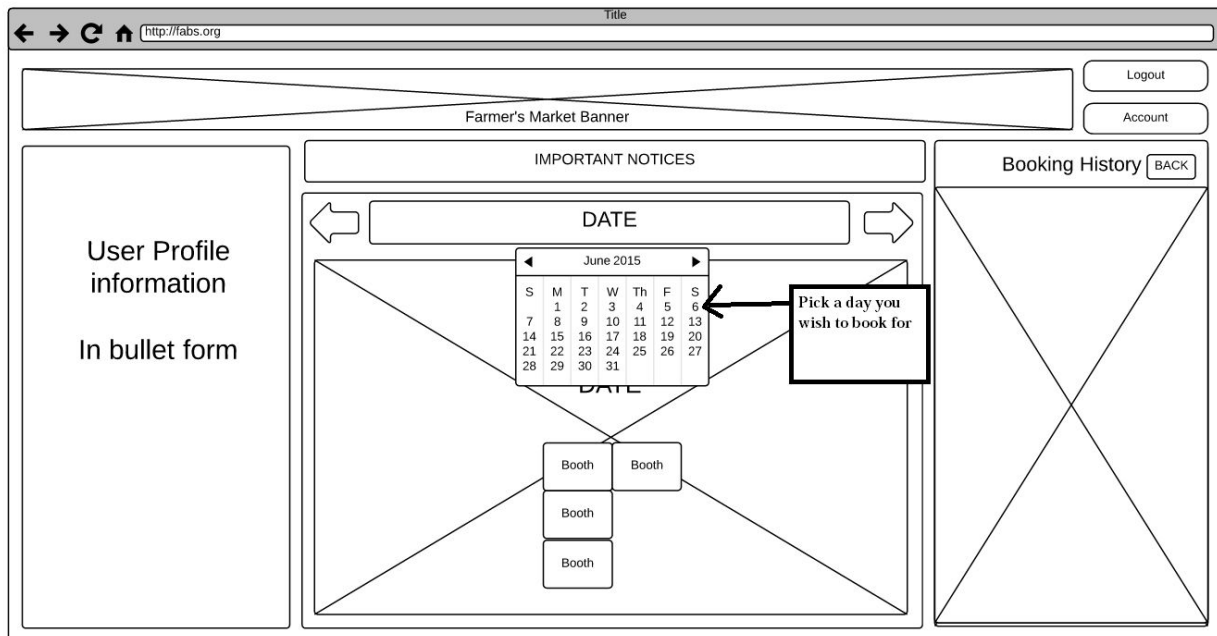
The screenshot shows a web browser window with the address bar displaying "http://fabs.org". The page title is "Register Your Account". It contains a vertical stack of input fields: "Market Booth Name:", "login name:", "password:", "confirm password:", "email:", "confirm email:", and "phone number:". To the right of the "Market Booth Name:" field is a callout box stating "User will fill all fields with their personal information" with an arrow pointing to the field. To the right of the remaining six fields are five upward-pointing arrows. At the bottom right are two buttons: "Back" and "Register". A callout box with the text "After filling out all required fields, user will click register" has an arrow pointing down to the "Register" button.

After filling out all required fields and clicking "register", you will be prompted to the home page of FaBS.

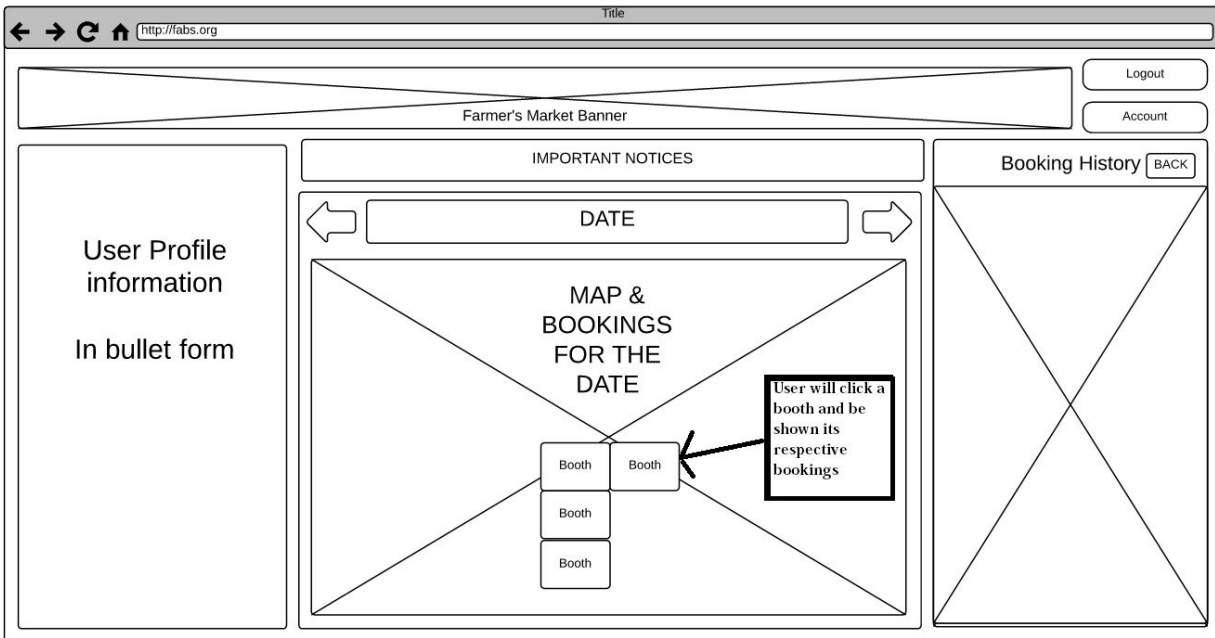
## Use Case 2: Booking a booth



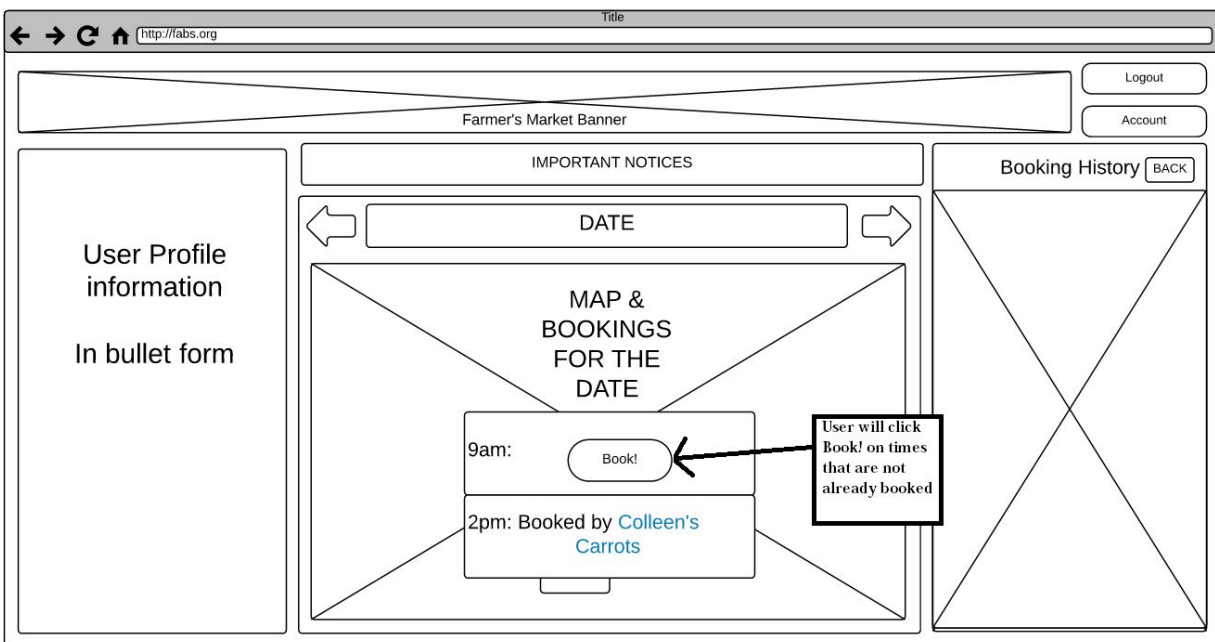
Assumed the user has logged in, the user will be shown the following screen. Users will be shown the bookings for the current day, but can change the day by clicking the arrow keys located above the map, or by the calendar that is shown after clicking on the date.\



After clicking on the date, the calendar will be displayed. Clicking on a day will enable you to choose what day the map will show.



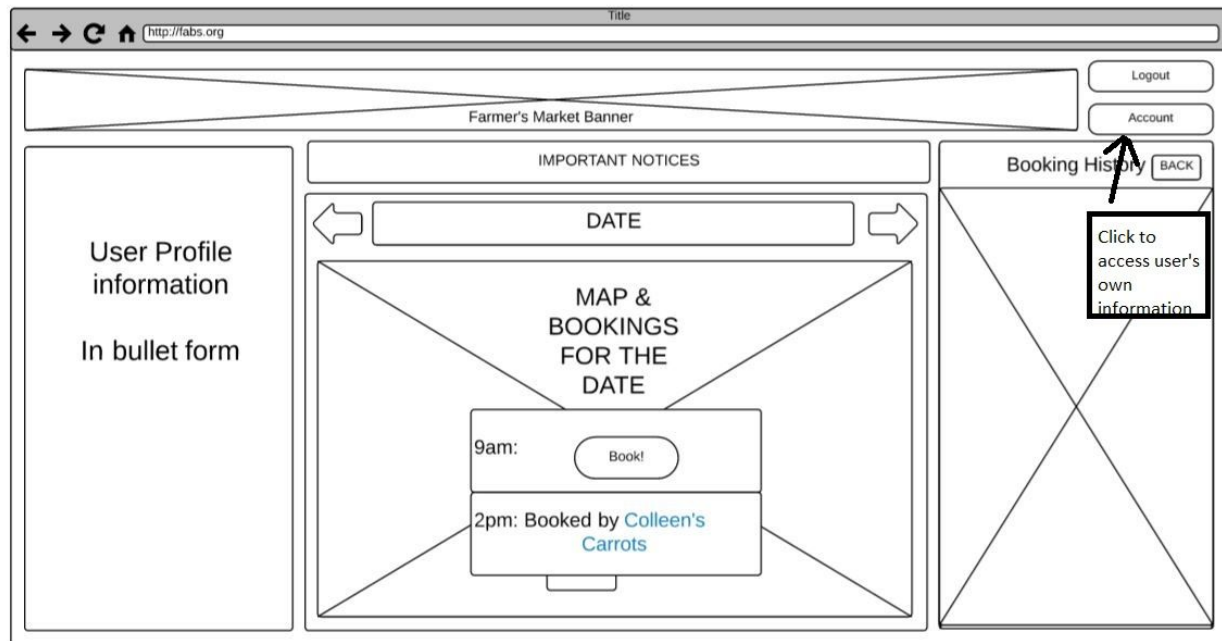
An updated map with different booth reservations will be shown. Clicking on a booth will bring you to the next screen.



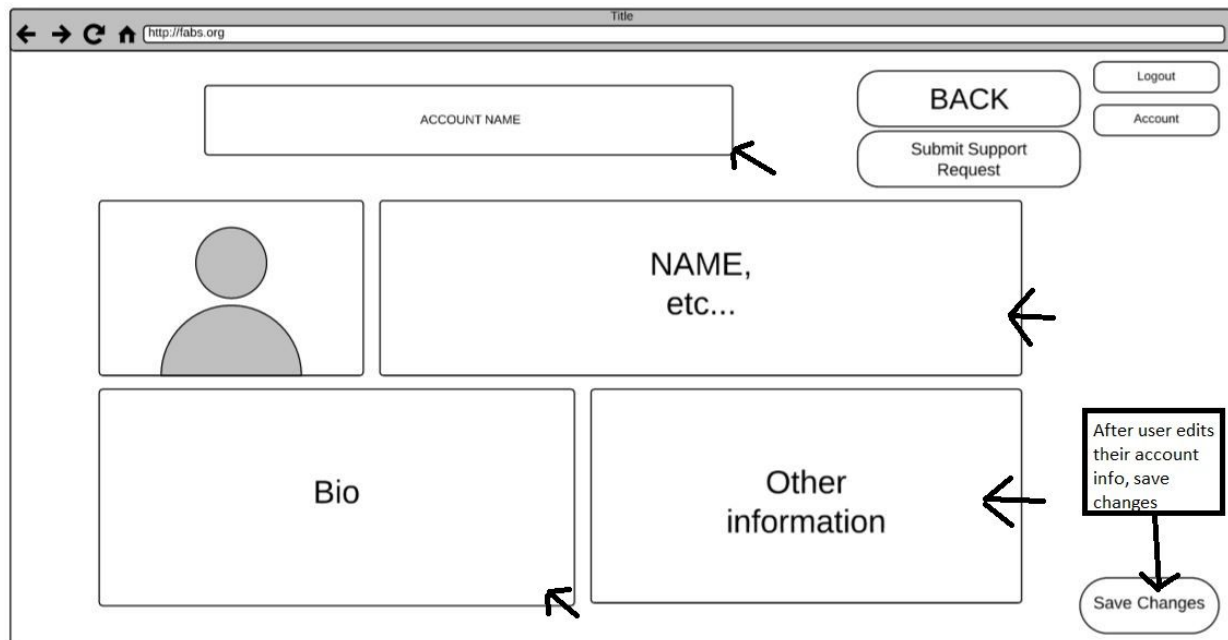
After clicking a booth, a lightbox will appear displaying that booth's bookings on the specified day. If there is an available slot, the "Book!" button will be displayed and the user is able to book that time slot.



### Use Case 3: User edits profile details



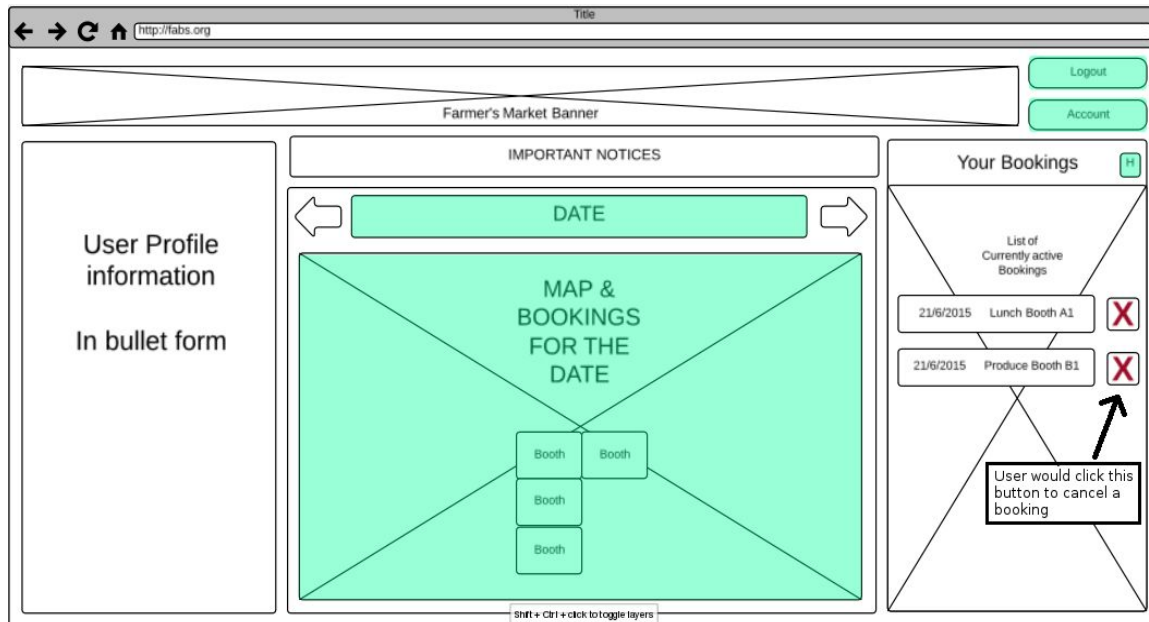
Assuming the user has logged in, the user will be shown the main screen. To edit their information, they are to click the “account” button to view their personal account information.



User will then be shown the screen of their user profile. The user is able to change what is already present to what they want. After the user is finished editing, they must click “Save Changes” in order for their changes to be saved.

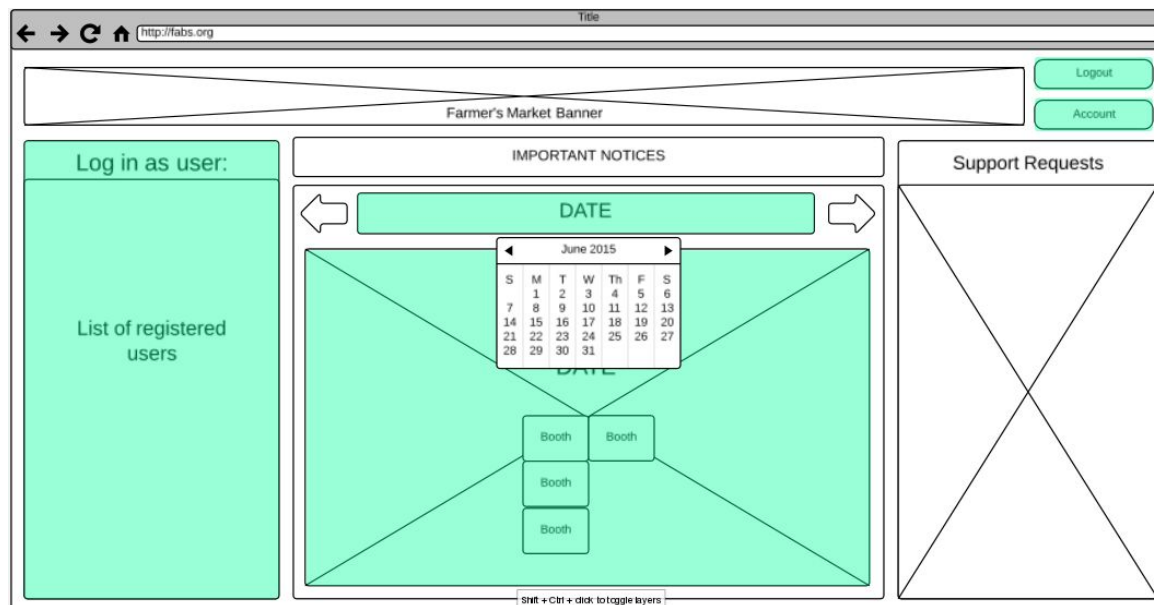
## Use Case 4: Cancelling a booking

### Use Case 4.1: Vendor Booking Cancellation:



This is the default UI view for a vendor when they are logged into the system. In order to execute this use case, the user will have to navigate to a particular booking under the “Your Bookings” section, and click on the red cross button to confirm cancellation. This would then remove the item from the list and register a cancellation within the back end of the system.

### Use Case 4.2 Administrator/System Operator Booking Cancellation:



This is the default view for an administrator when they log into the system. To cancel a booking, administrators will have to pick a date and then click on a booth to view booth bookings.

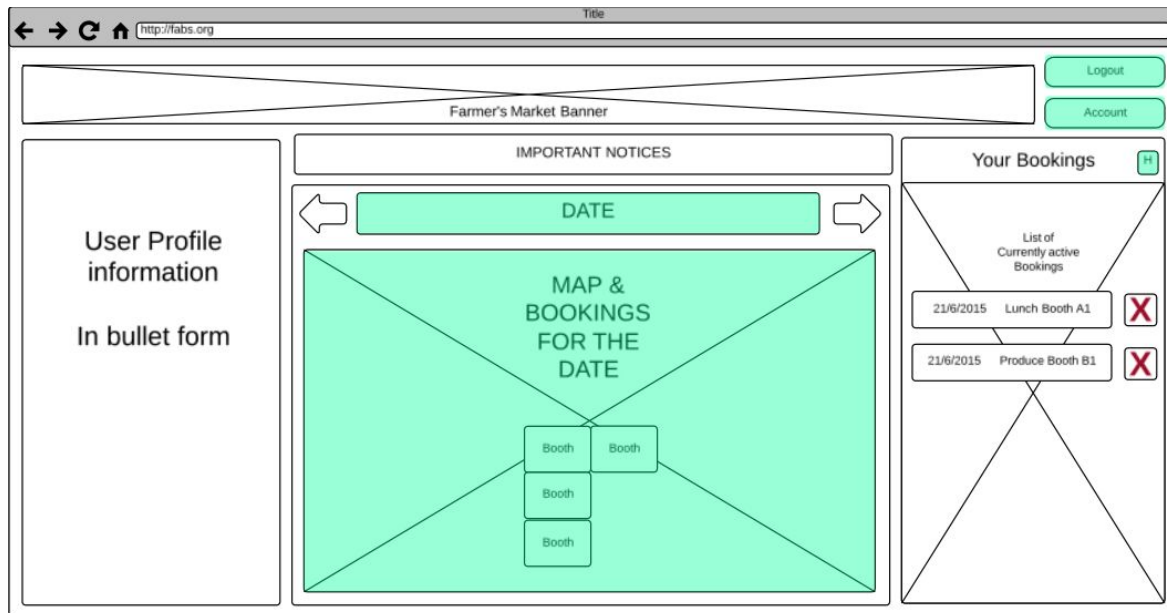
The screenshot shows the default administrator view. At the top is a navigation bar with a "Logout" button and an "Account" button. Below this is a "Farmer's Market Banner". The main content area is divided into three sections. On the left is a "Log in as user:" section with a "List of registered users" placeholder. In the center is an "IMPORTANT NOTICES" section with a "DATE" selector and a calendar for June 2015. Below the calendar is a booking confirmation box showing "9am:" with a "Book!" button and "2pm: Booked by Colleen's Carrots" with a "Cancel" button. On the right is a "Support Requests" section with a placeholder.

After being prompted with more information, they can then click the "cancel" button to confirm cancellation. Administrators can also cancel bookings by logging into the system as a registered user by clicking on an item from the list called "Log in as user" on the left hand section of the main page. This will lead them to the following page.

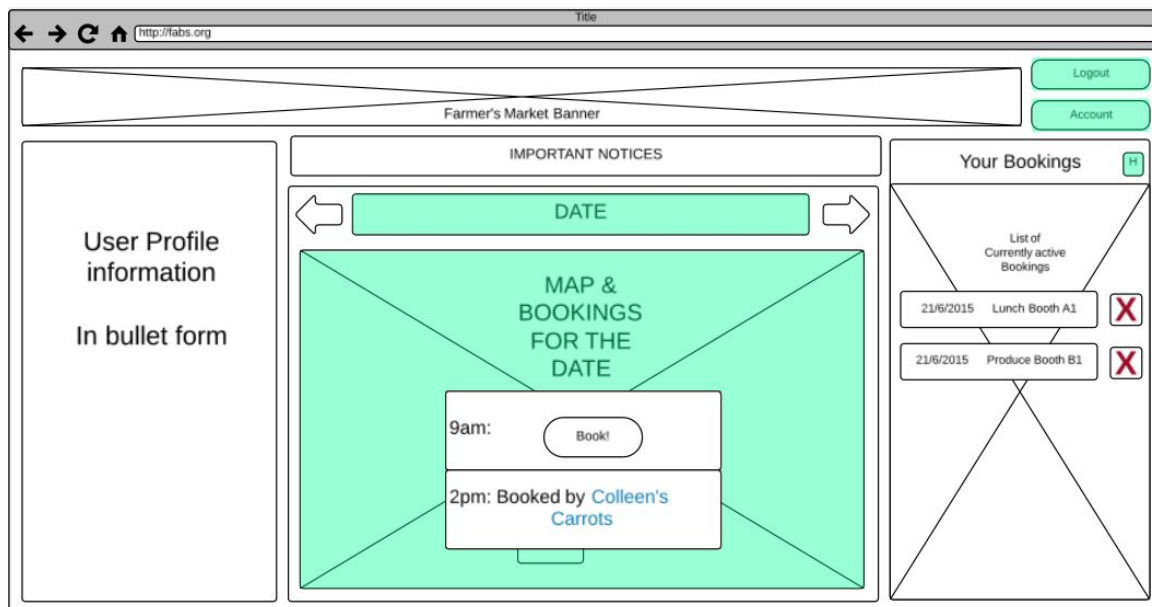
The screenshot shows the user profile view after logging in as a registered user. The navigation bar remains the same. The "Farmer's Market Banner" is still present. The left section now has a "Return to Admin View" button and a "User Profile information" section with a "In bullet form" placeholder. The center "IMPORTANT NOTICES" section is identical to the previous view, showing the "DATE" selector, the June 2015 calendar, and the booking confirmation box for "9am:" and "2pm: Booked by Colleen's Carrots". The right section now has a "Your Bookings" section with a placeholder.

They can then cancel bookings the same way any regular vendor can as described in Use Case 4.1.

## Use Case 5: User Viewing a Market Schedule

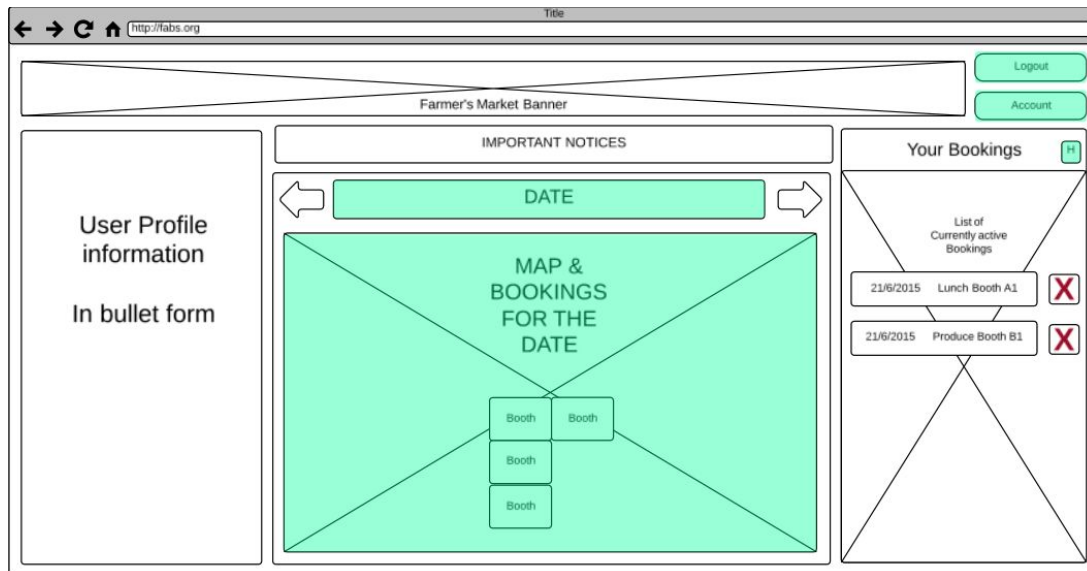


The map will display color coded booths that signify a booth's booking status on a given day. The user can then click on a booth according to their preferences and view further information. To view the market schedule of a different day, the user can pick a different date using the DATE section as shown in Use Case 4.2.

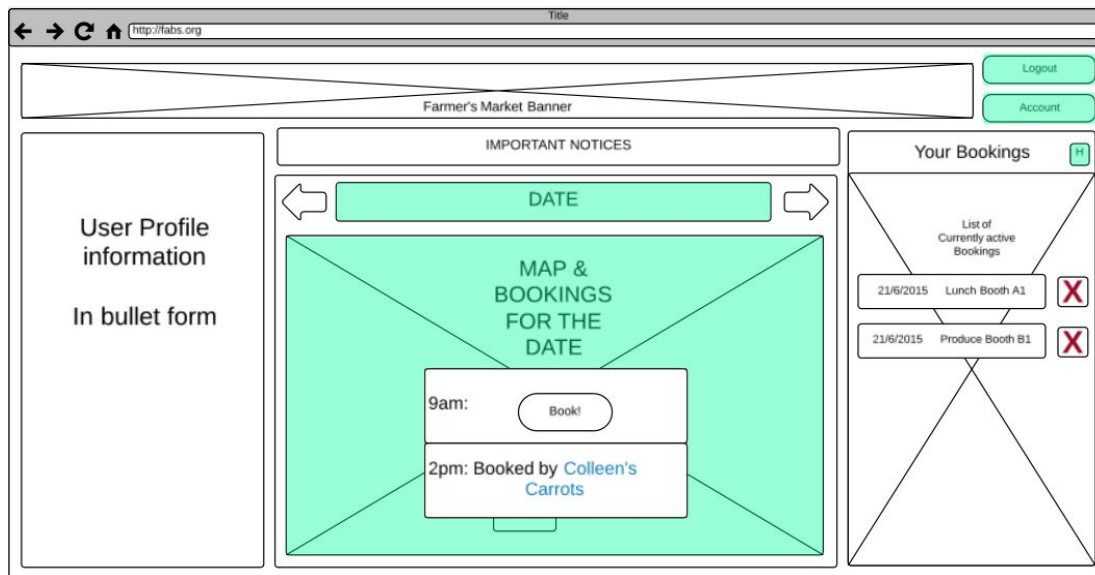


As an example, the user can then either choose to book an empty slot, or view more information about the vendor booking the 2pm slot by following the highlighted link.

## Use Case 6: User Views Other Booth's Information



To know more about a vendor occupying a booth, a user will have to navigate to a particular booth from the map and click on it.



To know more about the vendor, they can then click on the link, which in this case would be "Colleen's Carrots".

← → ↻ ⌂

http://fabs.org

Title

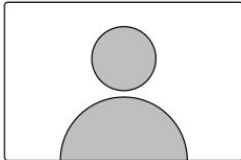
ACCOUNT NAME

BACK

Submit Support Request

Logout

Account



NAME,  
etc...

Bio

Other  
information

The user will then be redirected to the specific vendor profile. This will hold all relevant information that the vendor wants to share with the public including their specialties, biography, location, etc.

## Use Case 7: System Operator Creates New Administrator

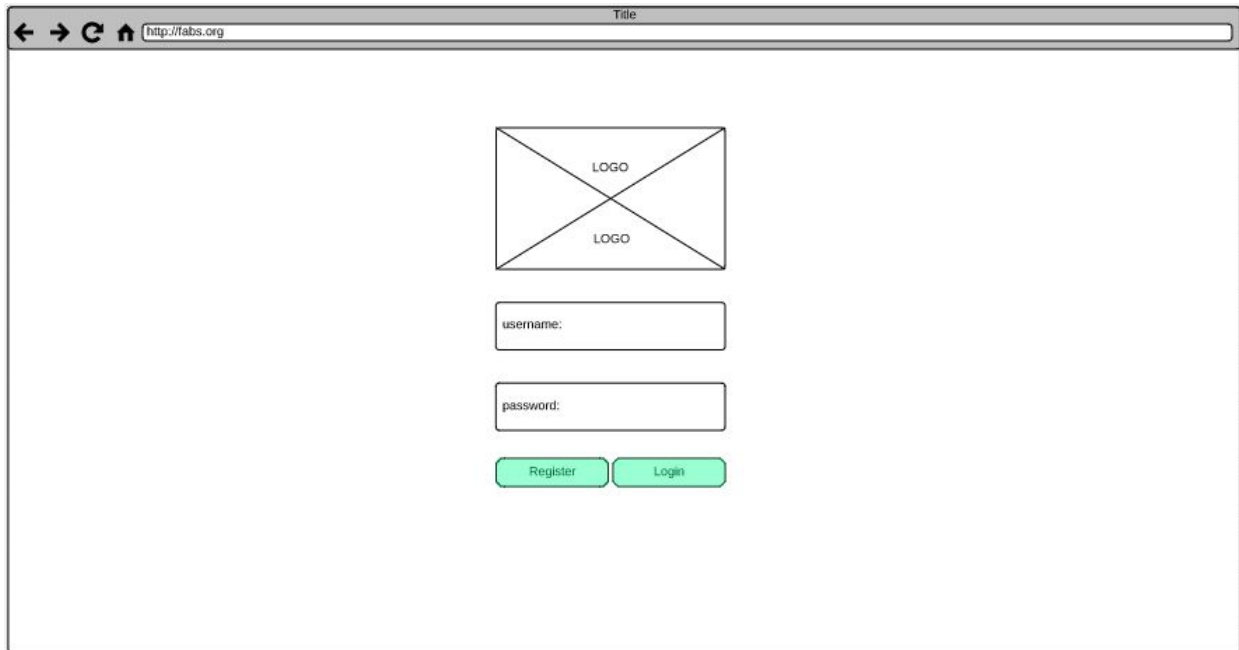
The screenshot shows a web browser window with the URL <http://fabs.org>. The page features a navigation bar with a "Farmer's Market Banner" and buttons for "Logout" and "Account". Below the banner, there is a "Log in as user:" section on the left, a central "IMPORTANT NOTICES" section, and a "Register New Administrator" button on the right. The "Log in as user:" section contains a "List of registered users" area. The "IMPORTANT NOTICES" section includes a "DATE" input field and a "MAP & BOOKINGS FOR THE DATE" section with four "Booth" labels. The "Register New Administrator" button is located in the top right corner. Below the button is a "Support Requests" section.

The system operator sees this screen when logged in. To create a new administrator, the system operator clicks on the button “Register New Administrator.” The system operator will then be taken to the following screen:

The screenshot shows a web browser window with the URL <http://fabs.org>. The page displays the "Register New User" form. The form includes input fields for "username:", "password:", "email:", and "phone number:". There are also input fields for "User Type: (admin or vendor)", "confirm password:", and "confirm email:". At the bottom of the form, there are two buttons: "Back" and "Register".

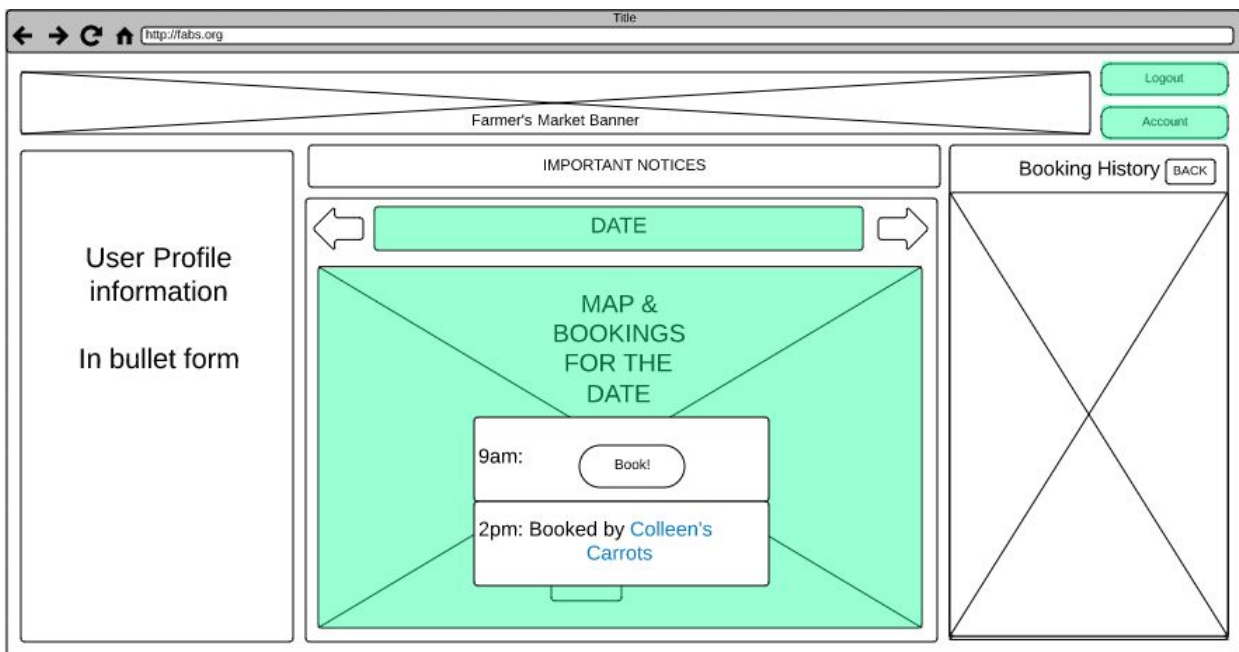
The system operator will enter administrator's' information, then click the button “Register.” Only a system operator can successfully complete registration with “admin” entered into the User Type field. A new administrator will then be added to the database.

## Use Case 8: Login to the System



A browser window titled "Title" with the address bar showing "http://fabs.org". The main content area features a placeholder for a logo (a rectangle with an 'X' and the word "LOGO" twice). Below the logo are two text input fields labeled "username:" and "password:". At the bottom are two green buttons labeled "Register" and "Login".

When a user visits the website, the initial screen is the log in screen. If the user does not have an account, the user must click on the “Register” button to create an account. Otherwise the user will enter their username and password in the corresponding textbox fields, and then click the button “Login.” The main pages that appear after login will correspond to the type of user: vendor, administrator, or system operator.

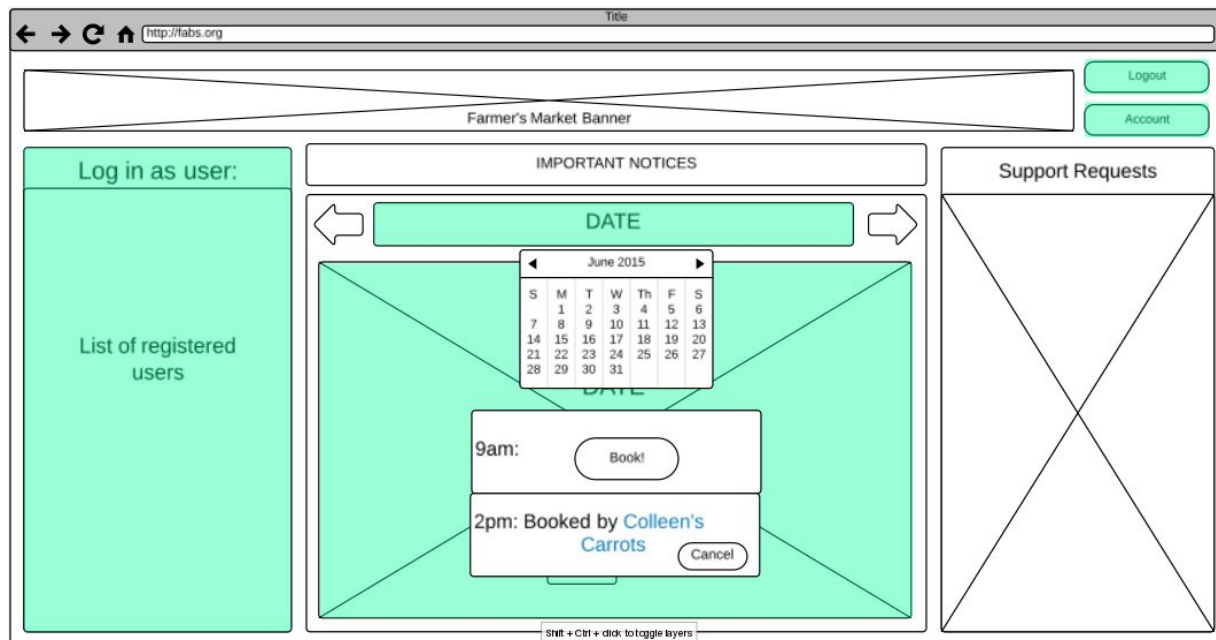


A browser window titled "Title" with the address bar showing "http://fabs.org". The dashboard layout includes:

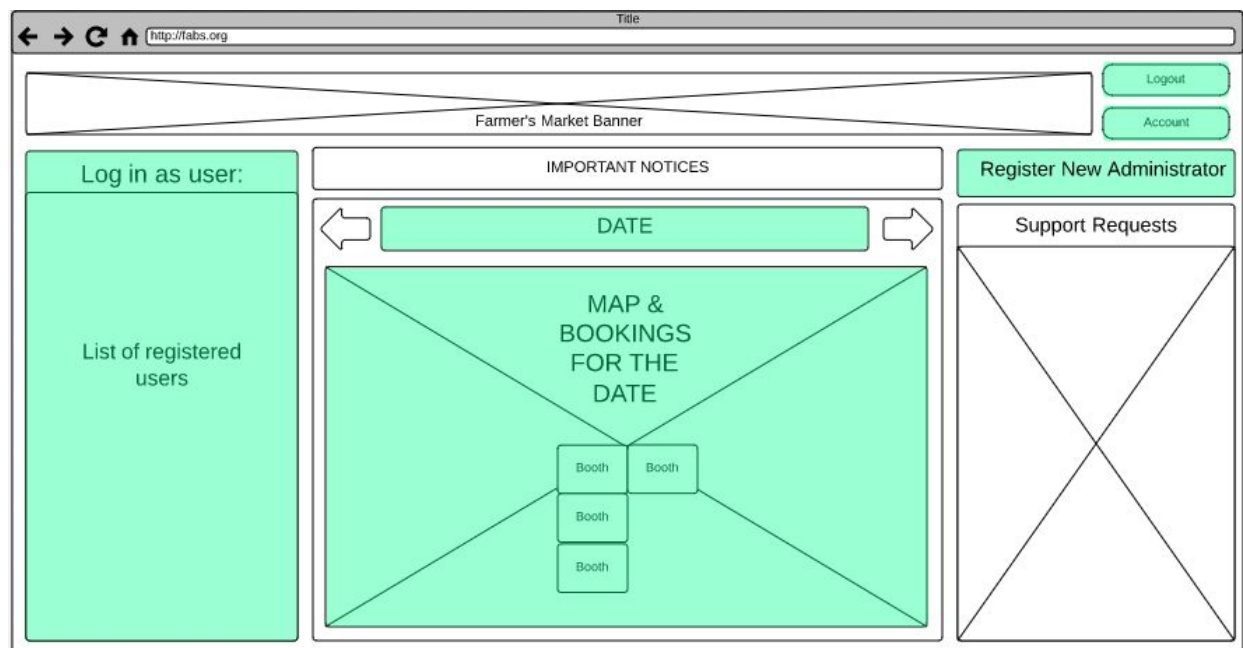
- Header:** A "Farmer's Market Banner" placeholder and two green buttons labeled "Logout" and "Account".
- Left Sidebar:** A box labeled "User Profile information" with the text "In bullet form".
- Center Content:**
  - A section titled "IMPORTANT NOTICES" with a green bar labeled "DATE" and left/right navigation arrows.
  - A large green area titled "MAP & BOOKINGS FOR THE DATE" containing a booking list:
    - 9am: [Book! button]
    - 2pm: Booked by Colleen's Carrots
- Right Sidebar:** A section titled "Booking History" with a "BACK" button and a placeholder for the history content.



The above screen shows the vendor main page.

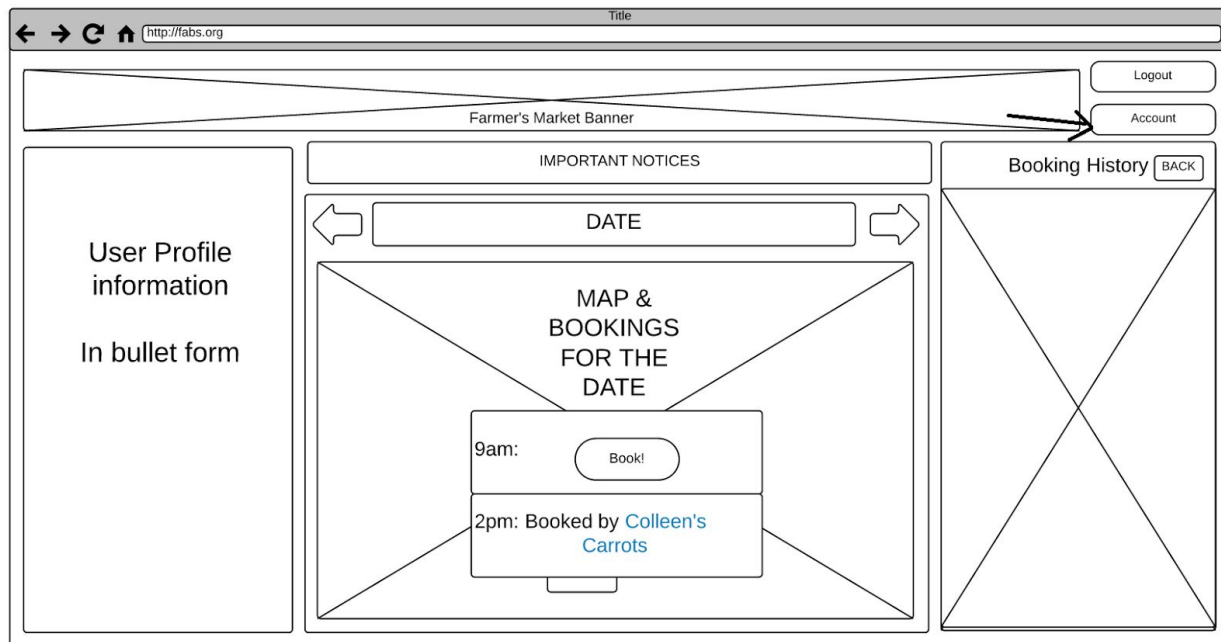


The above screen shows the administrator main page.

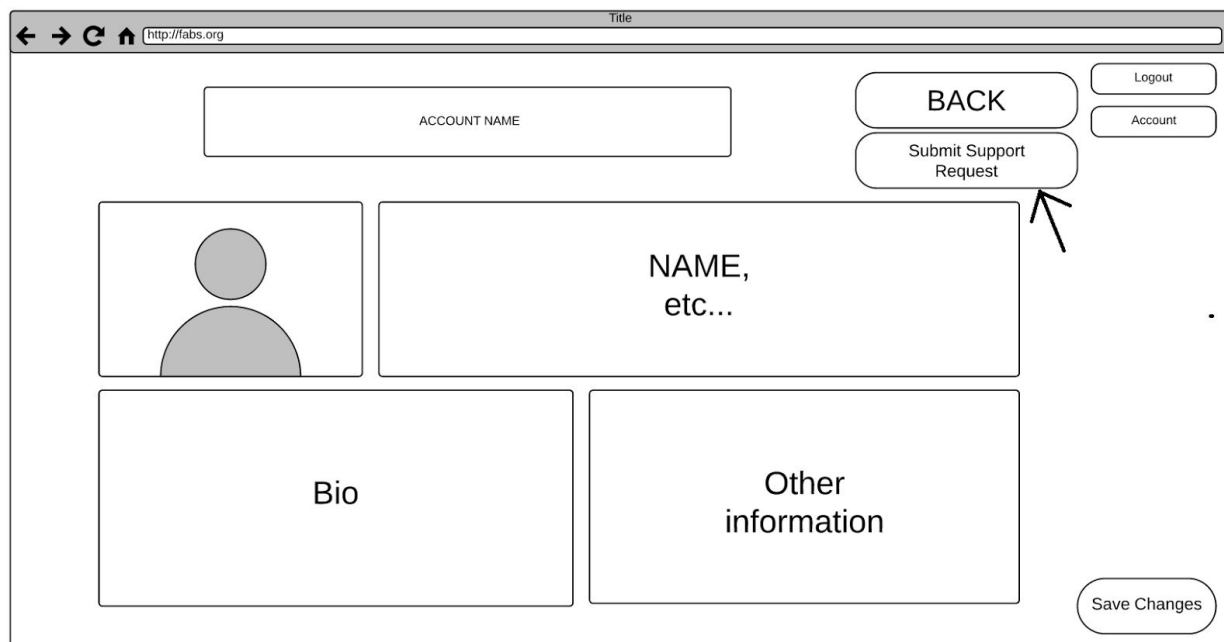


The above screen shows the system operator main page.

## Use Case 9: Create Support Request



After logging in, to submit a support request, the user will first have to click on the “Account” button to be taken to their account information.



User will then be shown the above screen. They will see all their account information, and are able to submit a support request by clicking the shown button.

← → ↻ ⌂ http://fabs.org Title

User will specify subject of request

Subject:

User will enter detailed information on request

Input box

User will finish their submission by clicking send

BACK

Logout

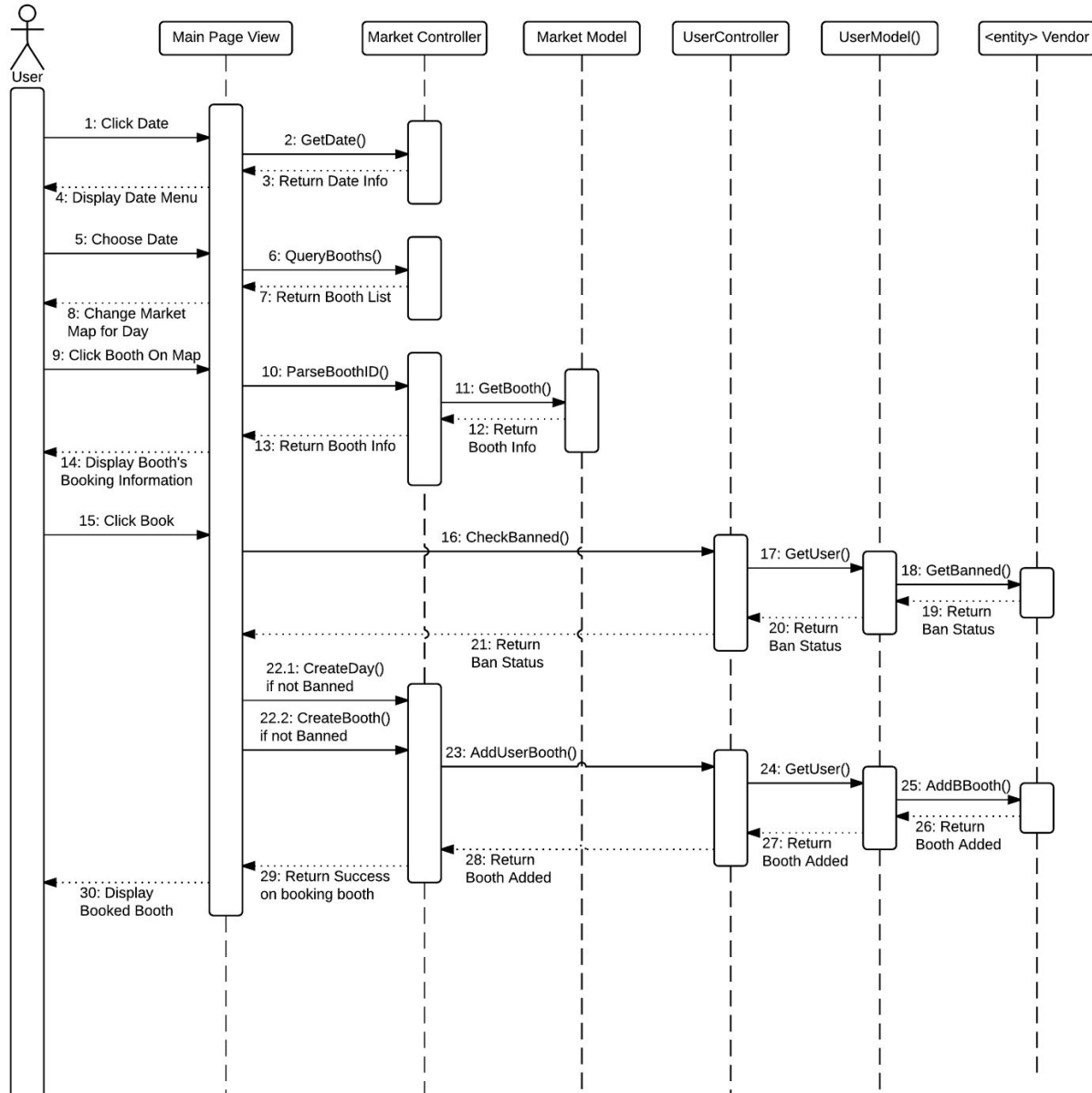
Account

SEND

User will fill in the text boxes shown. The subject will be the first thing shown to an admin. The input box is where the vendor will specify in detail their request. To complete the request, the user will click send.

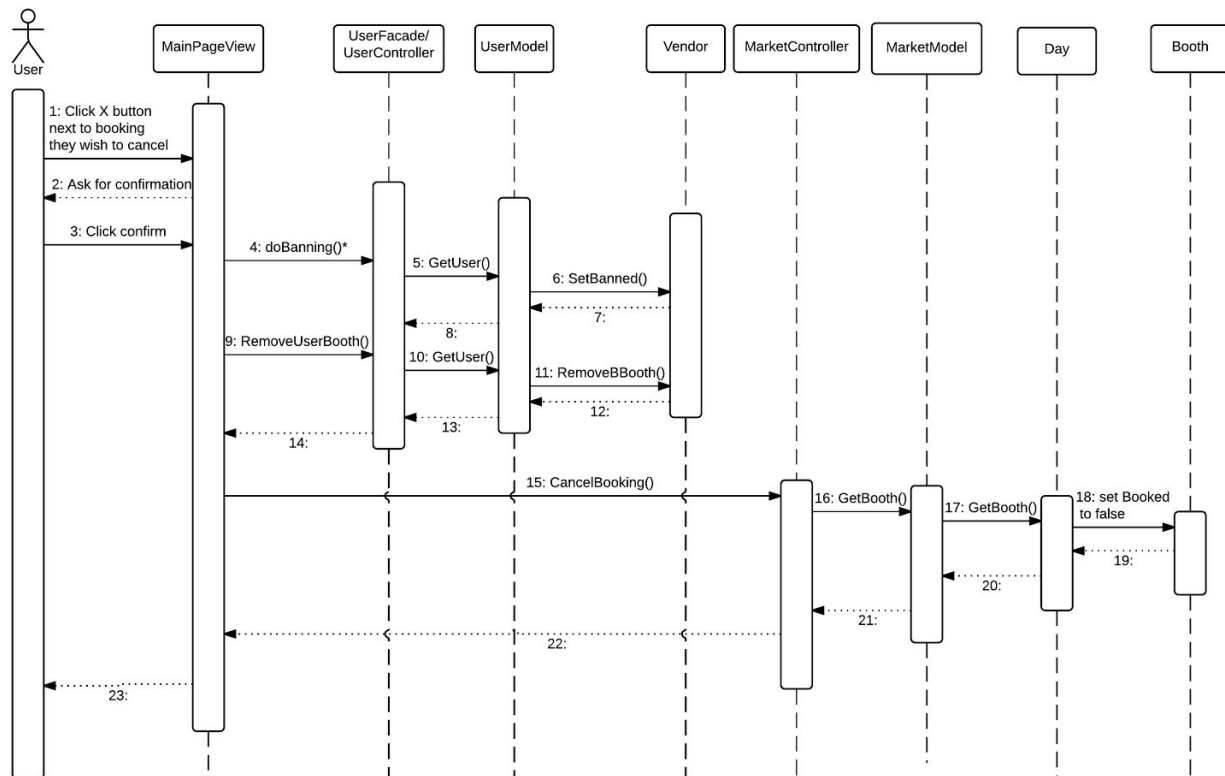
## 4.0 Sequence Diagrams

### 4.1 Booking A Booth



After the user has selected a booth to book, the system checks if the user is under the effect of a 48 hour ban. If they are, the system stops the booking process, else it continues and creates the necessary entities to then provide feedback to the user that they have booked the booth.

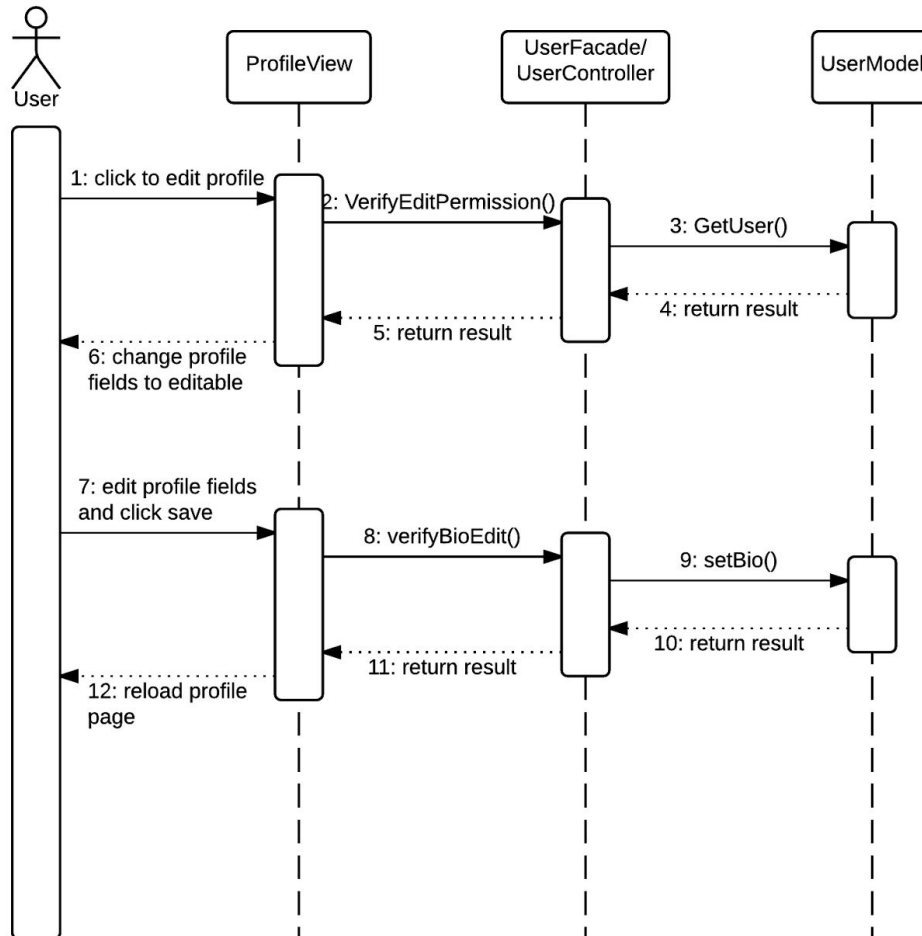
## 4.2 Cancelling A Booking



User Clicks 'X' next to the booking they want to cancel, then confirm they do want to cancel it. Now the system checks if canceling at this time should result in banning for 48 hours. If it does, it will get the user, and use the setBanned function to ban them for 48 hours. Next it will remove the booth from the users list of currently booked booths. Finally it will change the status of 'booked' in the booth entity to false. The canceling process is complete and control goes back to the user.

\* = the path is conditional

### 4.3 User Edits Profile Details



In this sequence the `UserFacade` merely calls functions in `UserController`. As such, the two classes have been collapsed into one node for the sake of readability.



