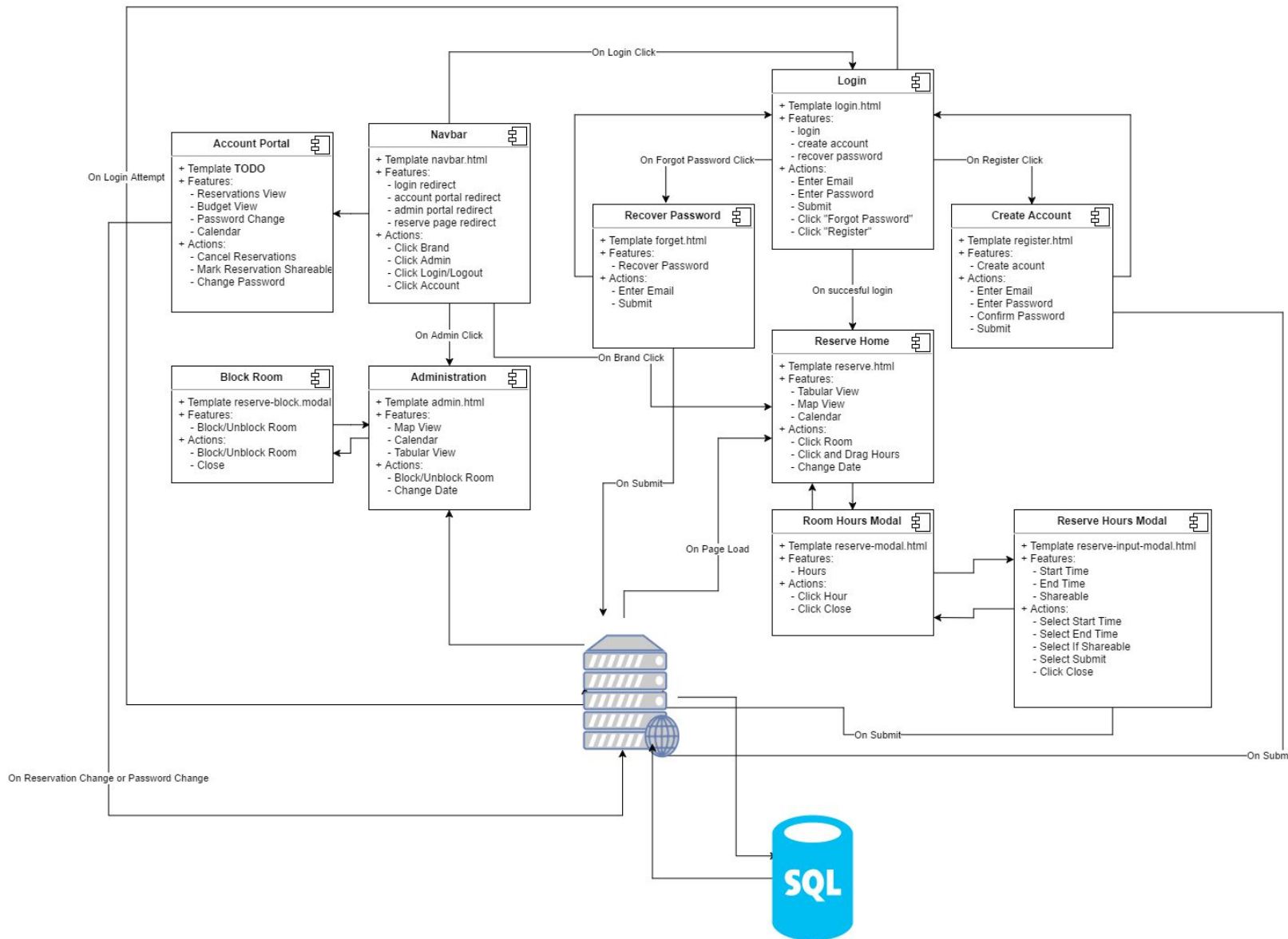


# Team 11: Defect Log

Pedro Del Moral Lopez, Sam Fellers, Ben Huemann, Michael Pike, Kaiwen Wei

# Design Description



## Design Inspection Defect Log

**Product:** RSVP: Campus Room Reservation System  
**Date:** 02/07/2017  
**Author:** RSVP Development team  
**Team Moderator:** Pedro Del Moral Lopez  
**Inspectors:** Sam Fellers, Ben Huemenn, Michael Pike, Kaiwen Wei  
**Recorder:** Sam Fellers

Defect #	Description	Severity	How Corrected
00	Reservation Times modal should be loaded before DOM objects.	3	Add as an angular directive under the reservation controller to make use of the "pre" load functionality
01	Calendar allowed the user to select dates that had already passed to attempt to reserve rooms.	4	Switched from a 3rd party DateTimePicker to the official Bootstrap DatePicker.
02	Room only has a single attribute attached to it, as well as room reservations.	4	Make reservations it's own table. For easy access when requesting multiple rooms' data.
03	Create Account input fields should display whether or not the current input fits the regular expression for the field.	3	Use the Bootstrap email and password input fields model instead
04	Hashing passwords should be on server side to prevent end users from being able to see how to passwords are hashed or to inject javascript preventing them from being hashed	1	Send passwords in plaintext to the server side which will then run the hashing algorithm and authenticate/store.
05	Rather than attaching an hours<> list to a room, which contains fields for whether the room is reserved or not for that hour, by who, how, etc. Have a separate registration data structure which only accounts for when rooms are reserved in order to save space in the database.	2	Add a separate reservations database.

06	Reservations should have a separate field for identifying individual reservation fields rather than just looking them up by matching user and date-time's in the database.	3	Add a reservationID to the reservation database that auto-increments each time a reservation is made to allow for a single identifier when requesting for reservation deletion.
07	Should not have an API call to get a week's worth of roomData, frontend would look messy displaying a whole week of reservations for a room, since the rooms in Hicks are all basically the same end users probably won't be biased to a certain room.	4	Remove the API call from the backend, instead add a datepicker which will allow you to see a day's worth of data for all the rooms. Making it easier to find open slots of time.
08			

# Code Inspection Defect Log

**Product:** RSVP: Campus Room Reservation System  
**Date:** 02/07/2017  
**Author:** RSVP Development team

Defect #	Description	Severity	How Corrected
00	Confirm Password doesn't work. Accept input of different password. Should display message that the password should be the same.	1	<b>TODO:</b> Add matching functionality in the frontend code.
01	The field of 'Confirm Password' shows the actual password instead of dots.	2	Set type of input field to "password" as opposed to "email"
02	Login page accept invalid password. Should display a message that the password doesn't meet the password requirement.	1	<b>TODO:</b> Fix the Angular code. Make sure the only password matching the requirement will be sent to backend.
03	Directive variables are initially undefined since the directive compiles prior to the data being provided. Causes errors in the console from attempting to reference attributes of undefined	1	<b>TODO:</b> Check for undefined and cease manipulation if data is undefined .
04	There is no Angular method associated with "Save Changes" on "reserve-input-modal" for making a reservation.	1	<b>TODO:</b> Write method
05	There is no Angular method associated with the dropdown inputs or checkbox on "reserve-input-modal" for retrieving the selected value in order to make the reservation.	1	<b>TODO:</b> Write method
06	There is no Angular method associated with the tabular timetable directive for creating a reservation.	1	<b>TODO:</b> Write method
07	There is no Angular method associated with the	1	<b>TODO:</b> Write Method

	Bootstrap Datepicker for querying a new day's data from the database and displaying to the screen.		
08	Admin.html's map uses the same code as the reserve.html for non-blocked rooms, making it impossible to block new rooms, just unblock old ones.	1	<b>TODO:</b> Design new modal for blocking rooms, implement Angular method for updating room information.
09	Admin.html's unblocking method does not change database information and does not unhighlight the room after unblocking.	1	<b>TODO:</b> Rewrite code to update database information, force refresh of page.
10	Admin.html's unblocking method has erroneous alert message	3	<b>TODO:</b> update alert displayed on unblocking.
11	Imagemapster does not dynamically update rooms as they are blocked and unblocked	1	<b>TODO:</b> write method for pulling new data, integrate with ImageMapster compile cycle.
12	Admin.html is using an erroneous datetimepicker.	4	<b>TODO:</b> Switch datetimepicker to Bootstrap Datepicker
13	Reserve.html's footer is not properly placed on page.	4	<b>TODO:</b> Update CSS to move to correct position.
14	Navbar does not update "Login" to logout when user logs in.	3	<b>TODO:</b> Write account based navbar into angular controller.
15	Dummy roomsData json array is not 0 indexed causing us to subtract 1 from roomId's can cause an error if entering 0.	2	TODO: Add Checks above all functions making a reference to roomsData[id-1].
16	Furthermore roomsData being not zero indexed, shows us that major errors can arise when functions like unBlock and checkBlocked are passing a roomId.	3	Zero-Indexed all rooms data in database to combat issues we had with our dummy data.
17	Most functions in the frontend don't return anything, rather change the display in some way. This makes it hard to test against different scenerios.	2	TODO: Add return values to functions designed to change display features with verbose success or failure codes, for both user and tester.

# Unit Testing

## Reservation Module Unit Testing Defect Log

<b>Product:</b>	RSVP: Campus Room Reservation System
<b>Date:</b>	02/07/2017
<b>Author:</b>	RSVP Development team

Defect #	Description	Severity	How Corrected
00	Tested unblockRoom() with integers below 0 and above 20 (the number of rooms we currently have), code broke	2	Added in checks to test whether the input is smaller than 0 or greater than the number of rooms we have.
01	Tested checkBlocked() with integers below 0 and above 20 (the number of rooms we currently have), code broke	2	Added in checks to test whether the input is smaller than 0 or greater than the number of rooms we have.
02	Cannot check values inside a forEach within toggleShareable, cannot check to see if return true ever fires.	4	<b>TODO:</b> Assign a variable inside the loop and do checks outside the loop

\* All reservation automated tests were done in JSFiddle <<http://jsfiddle.net/2Ny8x/864/>>

### Module Description:

#### Overall Description:

Manipulation of room reservations through creation of reservations, toggling reservations as shareable, blocking rooms and unblocking rooms. This module is associated with the map and tabular view gui elements.

**Unit Tested Functions:**

unlockRoom():

Input: roomid (int)

Output: false: if input out of bounds or already unblocked

True: if blocked

blockRoom():

Input: roomid (int)

Output: false: if input out of bounds or already blocked

True: if unblocked

checkBlocked():

Input: roomid (int)

Output: false if input out of bounds or room is not blocked

True: if blocked

init():

Input: none

Output: None. Loads internatl variable.

toggleShareable():

Input: roomid (int)

Output: True if successfully toggled shareability

False if erroneous input or if room is blocked

validate():

Input: roomData (JSON), hour (int)

Output: true if hour is already taken by a reservation

False if hour is free

validateShareable():

Input: roomData (JSON), hour (int)

Output: true if hour is reserved but can be shared

False if hour is reserved and cannot be shared.

**Untestable Functions:**

openModal():

Input: DOM event

Output: None. Manipulates DOM. Opens Hours Modal.

Reason for not testing: DOM is inaccessible from Unit Testing Framework

mouseover():

Input: DOM event

Output: None. Manipulates DOM. Temporarily Highlights Room.

Reason for not testing: DOM is inaccessible from Unit Testing Framework

mouseleave():

Input: DOM event

Output: None. Manipulates DOM. Removes temporary highlight.

Reason for not testing: DOM is inaccessible from Unit Testing Framework



disableBlockedRooms():

Input: None

Output: None. Manipulates DOM. Disables hour selection for reserved hours.

Reason for not testing: DOM is inaccessible from Unit Testing Framework

openHours():

Input: DOM event, int

Output: None. Manipulates DOM. Opens Create Reservation Modal.

Reason for not testing: DOM is inaccessible from Unit Testing Framework

closeSecondModal():

Input: none

Output: None. Manipulates DOM. closes Create Reservation Modal.

Reason for not testing: DOM is inaccessible from Unit Testing Framework

## Database Modules Unit Testing Defect Log

### Connection Module

Module Description:

#### **Overall Description**

Handles routes, POST and GETS from the front end.

### API Module

Module Description:

#### **Overall Description**

Handles interaction between Connection module and the SQL database through an established API

Methods:

#### **Unit Tested Functions:**

emailExists()

Input: email, connection

Output: callback

usernameExists()

Input: username, connection

Output: callback

addAccount()

Input: email, username, password, connection, callback

Output: callback

authAccount()

Input: email, password, connection, callback

Output: callback

deleteAccount():

Input: email, password, connection, callback

Output: callback

error

addReservation():

Input: roomID, user, date, startTime,

endTime, shareable, connection, callback

Output: callback

**Product:** RSVP: Campus Room Reservation System

**Date:** 02/07/2017

**Author:** RSVP Development team

Defect #	Description	Severity	How Corrected
00			
01			
02			

# Untested Modules

## Login Module

### Module Description

Login related code such as authentication, redirection, and data validation.

### Reason for not testing:

Code not implemented

## Account Creation Module

### Module Description

Account Creation related actions such as email authentication, password creation and data validation

### Reason for not testing:

Code not implemented

## Account Portal Module

### Module Description

Account related actions such as reservation manipulation (cancellation and shareability toggling) and password changing.

### Reason for not testing:

Code and HTML not implemented

## Account Recovery Module

### Module Description

Recovery account by providing email associated with an account for a recovery email to be sent to.

Reason for not testing:

Code not implemented