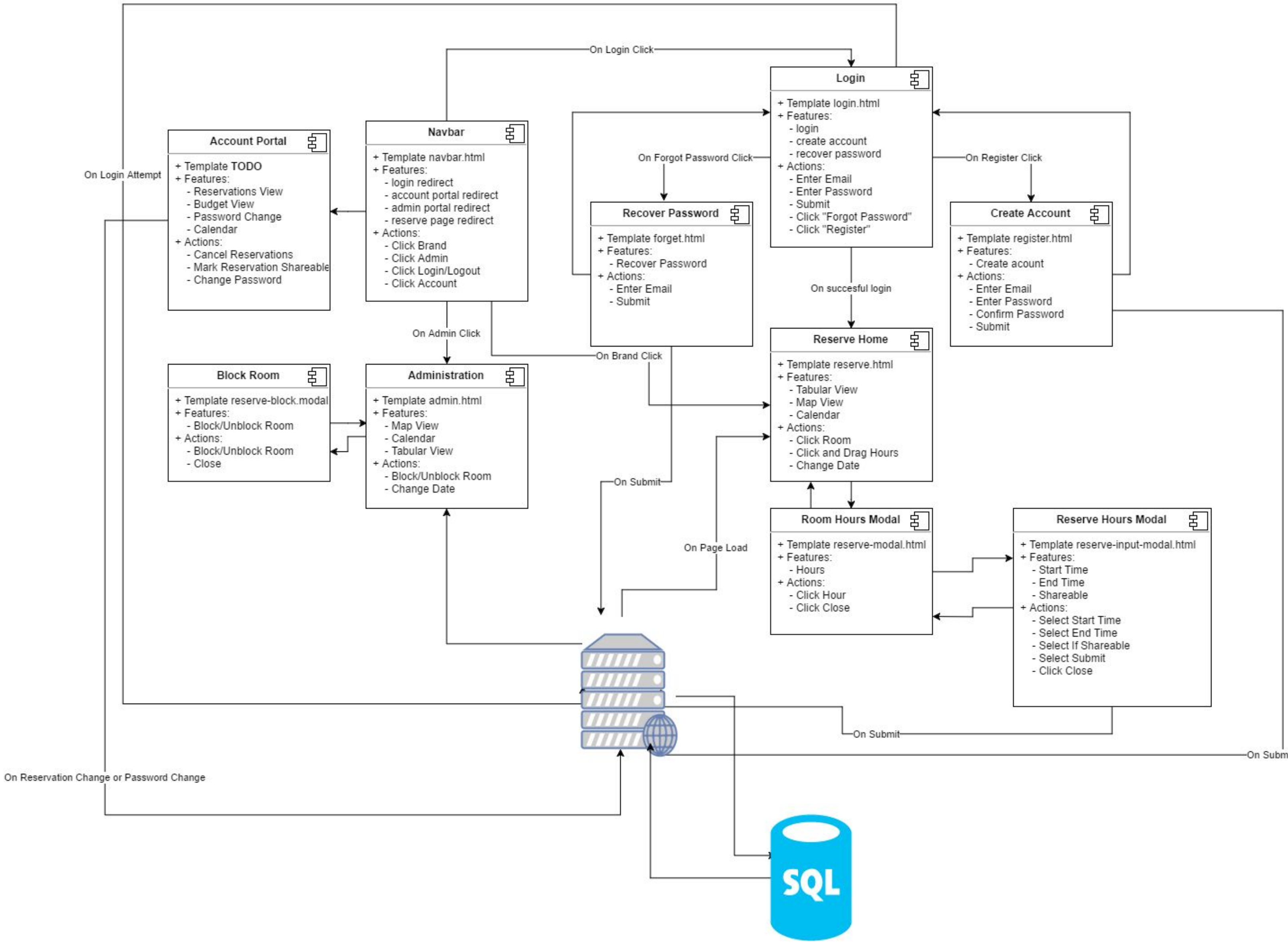
Team 11: Defect Log

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

Design Description



Account Portal
- + Template TODO
- + Features:
  - Reservations View
  - Budget View
  - Password Change
  - Calendar
- + Actions:
  - Cancel Reservations
  - Mark Reservation Shareable
  - Change Password

Navbar
- + Template navbar.html
- + Features:
  - login redirect
  - account portal redirect
  - admin portal redirect
  - reserve page redirect
- + Actions:
  - Click Brand
  - Click Admin
  - Click Login/Logout
  - Click Account

Login
- + Template login.html
- + Features:
  - login
  - create account
  - recover password
- + Actions:
  - Enter Email
  - Enter Password
  - Submit
  - Click "Forgot Password"
  - Click "Register"

Recover Password
- + Template forget.html
- + Features:
  - Recover Password
- + Actions:
  - Enter Email
  - Submit

Create Account
- + Template register.html
- + Features:
  - Create acount
- + Actions:
  - Enter Email
  - Enter Password
  - Confirm Password
  - Submit

Block Room
- + Template reserve-block.modal
- + Features:
  - Block/Unblock Room
- + Actions:
  - Block/Unblock Room
  - Close

Administration
- + Template admin.html
- + Features:
  - Map View
  - Calendar
  - Tabular View
- + Actions:
  - Block/Unblock Room
  - Change Date

Reserve Home
- + Template reserve.html
- + Features:
  - Tabular View
  - Map View
  - Calendar
- + Actions:
  - Click Room
  - Click and Drag Hours
  - Change Date

Room Hours Modal
- + Template reserve-modal.html
- + Features:
  - Hours
- + Actions:
  - Click Hour
  - Click Close

Reserve Hours Modal
- + Template reserve-input-modal.html
- + Features:
  - Start Time
  - End Time
  - Shareable
- + Actions:
  - Select Start Time
  - Select End Time
  - Select If Shareable
  - Select Submit
  - Click Close

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	**Fixed:** 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.	3	**Fixed:** 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.	3	**Fixed:** 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	**Fixed:** Use dthe 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	**Fixed:** 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	2	**Fixed:** Add a separate reservations database.

	registration data structure which only accounts for when rooms are reserved in order to save space in the 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	**Fixed:** 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.	3	**Fixed:** Requesting a week's worth of data is outside the scope of the deliverable and an unnecessary feature.
08	The navbar should display according to the user's status(admin or not, logged in or not).	3	**Fixed.** Added different html code for the navbar. Added Angular JS code to insert the code dynamically.
09	The Login button when user is currently logged in in the navbar should change to Logout. It should change the user status on the backend to Logged out instead of simply taking the user to the Login page.	3	**Fixed** by substituting different navbar based on session status.
10	Remember function on the login page not working.	3	**Fixed** through use of localStorage to store username and password then have it recall on page load
11	Reservation allowed even the user is not logged in.	2	**Fixed** through use of sessionStorage to require login to access page.
12	Parsing date is not working on the frontend.	3	**Fixed:** Modified the expected template to match with backend.
13	Click logout on the Account page shows up unknown account information (Possibly seeing other people's reservation)	2	**Fixed** through use of sessionStorage to require login to access page.

14	Blocking a room doesn't cancel reservations in that room.	2	**Fixed:**Reservations on a room that is going to be blocked are cancelled and an email is sent out informing the owner of the reservation
15	Admin isn’t given the option to block through a certain date	2	**Fixed:** This was determined to be outside the scope of the deliverable and not a necessary feature.
16	User cannot tell what the room is without clicking on it.	3	**Fixed:**The side table now provides all the necessary data for a room at a glance, including hours reserved and room name

Code Inspection Defect Log

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

Defect #	Description	Severity	How Corrected
00	Error handling when getting responses from the backend is not verbose and tends to be unique for each function call.	3	**Fixed:** All errors ported to standard new Error format
01	Index controller had become obsolete	3	**Fixed:** Removed index controller
02	Controller functions lengthy due to parsing dates	3	**Fixed:** Added new smaller, neater functions for doing parsing related tasks getToday() and parseDate()
03	adminOption() function does not have return values, making error checking difficult	3	**Fixed:** Added return values, true and false
04	adminOption() function makes calls to blockRoom and unblockRoom which do error checking themselves, makes for difficult to decipher error messages	3	**Fixed:** Rather than doing error checking for the same things in both, return the return value from blockRoom/unblockRoom in adminOptions
05	changePassword() function allows for passwords that do not meet password criteria	2	**Fixed:** Added checks for character length, uppercase, number, etc server side.
06	changePassword() allows undefined values to be entered into the changepass field	2	**Fixed:** Added checks for undefined password
07	authAccount can be bypassed by using SQL wildcards (%, _)	1	**Fixed:** Added checks in the frontend for these wildcards
08	Register regular expressions for valid emails and passwords are done in the html, can be changed to accept anything.	1	**Fixed:** Added the regular expression checks inside the back and front end javascript
09	Confirming passwords can be bypassed by manually sending the request to the api on the backend	3	**Fixed:** Added check to updateUserPassword function

10			
11			
12			
13			
14			
15			
16			

Unit Testing

Reservation Module Unit Testing Defect Log

Product:	RSVP: Campus Room Reservation System
Date:	02/07/2017
Author:	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	**Fixed:** 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	**Fixed:** 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.	3	**Fixed:** found erroneous reference to JSON values that don’t exist, changed to actual.
03	Tested validate() with an integer first parameter rather than a json, code broke.	3	**Fixed:** Added in a check to make sure room was defined after initialization.
04	Tested validateShareable() with an integer first parameter rather than a json, code broke.	3	**Fixed:** Added in a check to make sure room was defined after initialization.
05	Tested adminUpdateRooms() with an invalid date, no check on the frontend to allow this.	3	**Fixed:** Added in a check to validate year, not allowing users to request data for invalid years.
06	Tested parseDate() with invalid input, code broke, returning an array with partially undefined values	2	**Fixed**: Added in a check to make sure the input to parser is valid.
07	Tested parseDate() with error handling, however when date itself was undefined the code broke again.	2	**Fixed:** Added in a check to make sure date was defined before checking validity
08	Tested adminOption(), when given a room that was already blocked it returns false	2	**Fixed**: Was checking blocking manually instead of using the checkBlocked() function , manual checking is 0 indexed, checkBlocked is not.

* All reservation automated tests were done in JSFiddle, JSFiddle represents code prior to fixes of defects*
*function unit tests http://jsfiddle.net/sfellers/wydh1nwL/

*post request unit tests specifically for inputs which manipulate date in some fashion before sending to server http://jsfiddle.net/sfellers/hxdaoj8r/

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:

unblockRoom():

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 internal variable.

toggleShareable():

Input: reservationId (int), 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.

adminOption():

Input: roomIndex (int)

Output: true if block/unblockRoom() is returned a success

False if block/unblockRoom() fails.

changePassword():

Input: newPassword (String), confirmPassword (String)

Output: true if the passwords are the same.

parseDate():

Input: date (String) in mm/dd/yyyy form

Output: date (String) in yyyy-mm-dd form getToday():

Input: date (String) from getDate()

Output: date (String) in yyyy-mm-dd form

updateAdminRooms():

Input: date (String)

Output: returns true if the post request does not send back an err flag

share():

Input: click-event (changed to an id (int) for unit tests.)

Output: returns true if the post request does not send back an err flag

register():

Input: username (String), password(String), confirmpassword(String), email(String)

Output: returns false if password and confirm password arent the same, returns false if email is not valid

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

openAdminModal():

Input: none

Output: None. Manipulates DOM. opens admin modal for blocking rooms.

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

confirmLogin():

Input: Session information and localStorage

Output: None. Manipulates DOM. manipulates “remember-me” checkbox

Reason for not testing: LocalStorage and Session information could both be spoofed for testing purposes, but then all we would be testing is if an if-statement when given a true value returns true.

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
getRoomBlockedStatus():
        Input: roomID, connection, callback
        Output: callback
updateAccountPassword():
        Input: username, oldPassword, newPassword, connection, callback
        Output: callback
setRoomBlockedStatus():
        Input: roomID, status, connection, callback
        Output: callback
setReservationSharable():
        Input: roomID, status, connection, callback
        Output: callback
getAllRooms():
        Input: roomID, status, connection, callback
        Output: callback
```

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

Defect #	Description	Severity	How Corrected
00	Wrong error callback description is sent on trying to add an existing username, meaning the logic checks are not firing correctly	1	**Fixed** (email actually wasn't in the database)
01	Time is not correctly being saved (saved as NaN) on Reservation Test 1	2	**Fixed (**Time variable was not properly passed)
02	Callback messages are defaults currently	4	**Fixed (Callbacks were**

			reworked)
03	updateAccountPassword allows change to non-conformant password	3	**Fixed** (added check to function before SQL call)
04	getAllRooms returns all rooms, once for each room in the database	3	**TODO** (change api method to only get rooms once)
05	updateAccountPassword returns no error when given invalid credentials	2	**Fixed** (added errors to return value for api call)

Unit testing has been automated using a bash script and a set of separated unit tests in the Unit Test directory of the web app.