Software Requirements Specification & Design

for

Study Room Availability

Version 1.3 approved

Prepared for Nasseef Abukamail

Prepared by Fadi Alrabadi

Noah Glazier

Alex Neargarder

Zelin Zhang

Programming Paladins

April 26th, 2022

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 1

1.4 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 1

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 2

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 3

4.1 Campus Study Room List 3

4.2 Check Availability 4

4.3 Favorite Rooms 4

4.4 Reserve a Room 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 6

6. Design 7

6.1 UML Diagram 7

6.2 ER Diagram 7

6.3 Use Cases Diagram 7

6.4 Activity Diagram 7

6.5 Technologies Used 8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Document created | October 5th, 2021 |  | 1.0 |
| Document updates | November 2nd, 2021 | Feedback received after first submission. | 1.1 |
| Document updates | January 28th, 2022 | Changes to the website layout requested by the client. | 1.2 |
| Document updates | April 26th, 2022 | Changes to the design and specifications | 1.3 |

# Introduction

## Purpose

This SRS will specify the requirements for the Study Room Availability project, which is being developed by the group “Programming Paladins”. This SRS will cover the entire scope of this project.

## Intended Audience and Reading Suggestions

The design and user manual are intended to teach users and testers how to use the application. This document is also intended for developers and staff to be used as a guide in the creation and layout of the application.

## Product Scope

This software is going to provide a tool for students who are looking for an available room to study in. It will have support for adding a list of all the indoor study rooms around Ohio University and display the number of people currently in any given one. The goal is to get the software working with at least one room. It should be able to read information from sensors installed in that room, determine how many people are in that room, and then update the website and app to display how many people are in that room.

## References

* [Raspberry Pi User Guide (unca.edu)](https://www.cs.unca.edu/~bruce/Fall14/360/RPiUsersGuide.pdf)
* [Sensor Information and Schematic](https://robotic-controls.com/learn/sensors/pir-sensor-hc-sr501)
* [Tutorial for using Sensors with Raspberry Pi](https://dronebotworkshop.com/using-pir-sensors-with-arduino-raspberry-pi/)

# Overall Description

## Product Perspective

This product is designed to be a new, self-contained product. The idea for this product came from a professor at Ohio University who thought it would be a good idea to create a way for students to easily find study rooms that are unoccupied. In a way this product is part of a product family as there are several products being made currently that will have very similar functionality to it. There is also a product being made that provides a similar but distinct service by showing students how many people are in a professor’s office. This would allow them to know whether the professor is busy or available to talk without having to communicate with anyone.

## Product Functions

With this product, a user will be able to:

* View a list of all the study rooms on campus
* View a list of their personal favorite rooms
* View the status of any given room (Occupied / Vacant)

## User Classes and Characteristics

There will be two types of user classes within this product:

* A user class where any regular student will be able to create an account and use the application as it is intended to be used. The ability to favorite a room will mostly pertain to this user class.
* An administrator class where an administrator would be able to close and open rooms manually, in order to reserve rooms. As well as removing a room from the list and marking it unavailable on the application.

## Operating Environment

The software will be available to operate on android and iOS platforms as well as a web-based platform for which users who do not have a phone can visit. For hardware, we will be using a type of sensor which will detect whether a study room is empty or not, as well as how many occupants are in a study room, and will constantly update in real time and relay that information to the software.

## Design and Implementation Constraints

All testing of the Android application will be done through an emulator on a desktop, due to no developers owning Androids.

The application will be available to anybody with a Google account. Limiting access to Ohio University students and faculty would require access to the university’s login authentication.

## User Documentation

A user manual will be delivered along with the software.

## Assumptions and Dependencies

The sensor itself may not be able to tell us exactly how many people are in the room. Another issue with the sensor would be that it can only go off every 2.5 seconds compared to a much more expensive sensor. This would normally not be an issue but if we are counting the number of people then if two people were to walk in together then the sensor would only detect be tripped once.

# External Interface Requirements

## User Interfaces

The user interface will show a table with a list of all study rooms that the sensors are installed in. The table will include the room status (occupied or vacant). There will be a button for favoriting a room, and a button for favoriting a building. There will be an option to view your favorited rooms or buildings.

The admin user interface will include a button for editing room info and adding rooms.

## Hardware Interfaces

The supported devices will be phones running Android and phones running iOS. The supported web browsers will be Google Chrome, Firefox, Safari, and Microsoft Edge.

## Software Interfaces

The website and app will be connected to a database that contains the list of study rooms. The database will be accessed and updated when the sensors detect movement. The sensors will be connected to a Raspberry Pi, which will have a simple program running on it to receive the input from the sensor. When the sensors detect movement, the program will communicate with the web server so that the status of the room can be updated.

## Communications Interfaces

This project will be primarily using HTTPS for communication. The project features a website, so accessing that website will happen over HTTPS. The phone apps and Raspberry Pi will also need to communicate with the web server. This will happen over HTTPS as well.

# System Features

## Campus Study Room List

4.1.1 Description and Priority

This feature will allow users and administrators to view all of the study rooms on campus. This will be a list view that includes the room location and a picture of the room.

4.1.2 Stimulus/Response Sequences

The rooms will be listed when the application is opened.

4.1.3 Functional Requirements

A database will contain all of the rooms and their corresponding information. These rooms will be retrieved from the database and listed out within the application.

REQ-1: Working database containing room info

REQ-2: Allow program access to the database

## Check Availability

4.2.1 Description and Priority

A user or administrator will be able to check if a given room is occupied or vacant.

4.2.2 Stimulus/Response Sequences

When the user or administrator clicks on a room from the list, the room will show

up with its information and availability.

4.2.3 Functional Requirements

Infrared sensors connected to a Raspberry Pi will update room availability in the

database.

REQ-1: Create database that Raspberry Pi can update

REQ-2: Create GUI buttons for the rooms

## Favorite Rooms

4.2.1 Description and Priority

Users and administrators will be able to add rooms to a list of their favorites.

They will also be able to sort rooms by their favorites.

4.2.2 Stimulus/Response Sequences

When the user or administrator clicks on a room from the list, there will be another button (star) to add the room to your favorites.

4.2.3 Functional Requirements

Store users favorite rooms in a database. Create an option for sorting rooms

REQ-1: Create database to store room favorites

REQ-2: Create a GUI button for adding a room to favorites

REQ-3: Allow rooms to be sorted by favorites

## Reserve a Room

4.2.1 Description and Priority

A user or administrator will be able to reserve a room.

4.2.2 Stimulus/Response Sequences

When the administrator views a room from the list, there will be a button to reserve the room.

4.2.3 Functional Requirements

After clicking the button to reserve a room, the room will be marked as unavailable in the database and will show up as such on the application.

REQ-1: Create GUI button for reserving a room

# Other Nonfunctional Requirements

## Performance Requirements

Performance requirements for this application include:

* The product is required to update in real time once it receives data from the sensor in the room.
* Once the sensor goes off, the room will show as occupied on the mobile app and web app as well.
* Any reservations on the app that go through will automatically update the app and show that said room is unavailable at the time of the reservations.
* Favoriting a room would automatically save that room for the user and give the user the option to sort rooms by favorites.
* When a room is added or removed by an administrator, the application automatically adds or removes the room so that the users can or cannot interact with said room.

## Safety Requirements

To prevent an outsider from using our app for malicious purposes, we would require users to create an account using their school email address or give them the option to sign in with Google or any other platform that has this feature available such as Facebook or Apple.

## Security Requirements

The software will use Google login authentication and maybe other media login authentication as well as an option to create an account using your student email account which will receive a link to verify that the user is using his/her own account. The user information will be encrypted and protected in our database. Our main focus will be to protect user information from being hacked and used against users or sold without their knowledge. We will require the user to create a password that uses capital letters and symbols, to prevent them from being easily hacked.

## Software Quality Attributes

The product should be easy to use and navigate, as well as be available to anyone, no matter what kind of phone they have. This product should work on iOS devices, android devices and web browsers. It should make it so that the user does not have to waste time going into multiple buildings looking for a study room and should just tell them where an available one is located. It should update the availability of the study room in real-time so that the user can see if a room is no longer available or not.

## Business Rules

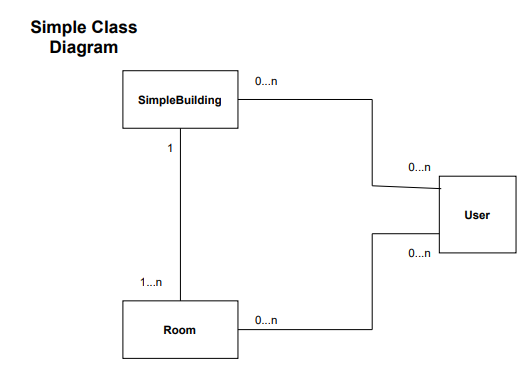
Administrators will have the ability to add or remove a room from the list of study rooms. Removing a room is done when that room is no longer used as a study room. This could be temporary or permanent. Adding a room is done when a closed study room opens again or if a new study room has been created. An administrator will also be able to accept or deny reservations of said room. When a user requests to reserve a room, the administrator is notified and will have the option to accept or deny the request.

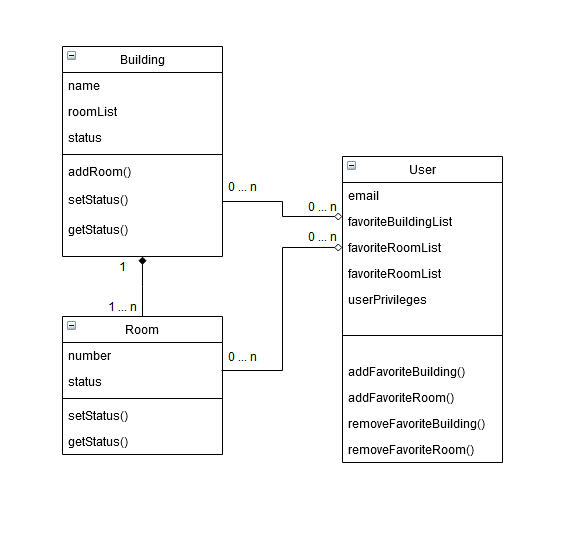
Users will have the option to perform a handful of functions on their app such as favorite a room and request to reserve a room at a specific time and date. A user will be able to favorite a room that they like to use, or favorite multiple rooms and when they open the app, they can select to view the availability of only their favorite rooms

# Design

## UML Diagram

The UML diagram below shows the basic class structures that our software will use.



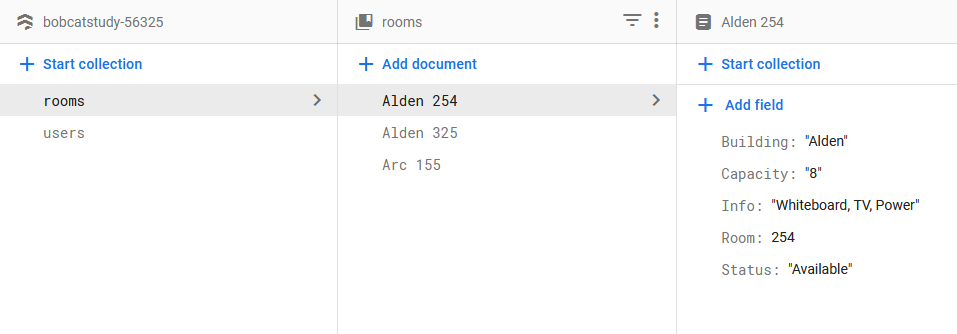


* The set status function in the building class will only be used in cases where the building is shut down. This field will only be able to be set by a user with administrator privileges.
* The userPrivileges attribute will indicate whether a given user is a normal user or an administrator. If they are an administrator, they will have the ability to override a room’s status and set the status of buildings.

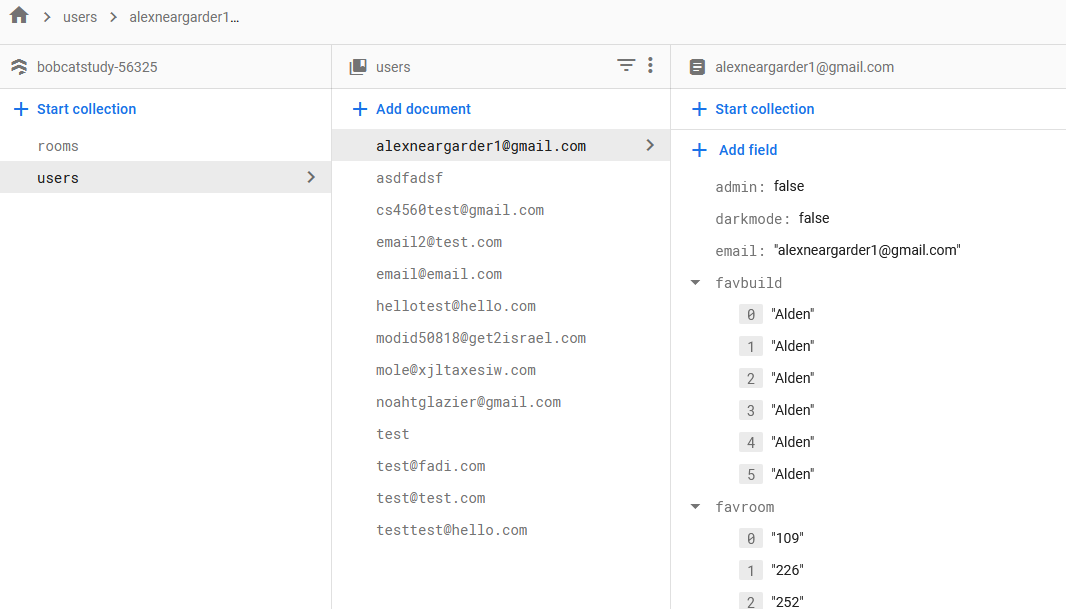
## ER Diagram

This diagram illustrates the structure of our databases. We have separate databases for storing the rooms and users.

The rooms database has an entry for each room, and each entry has the building, room number, capacity, room info, and room status.

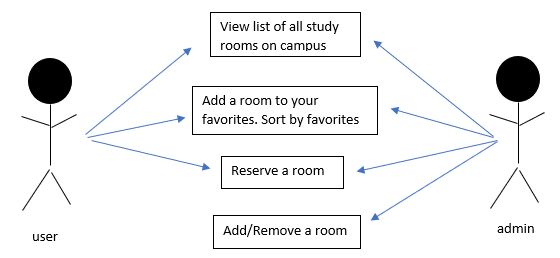


The users database has an entry for each user. Each entry contains the user’s email, favorite rooms, favorite buildings, and a variable to indicate whether they have dark mode enabled.

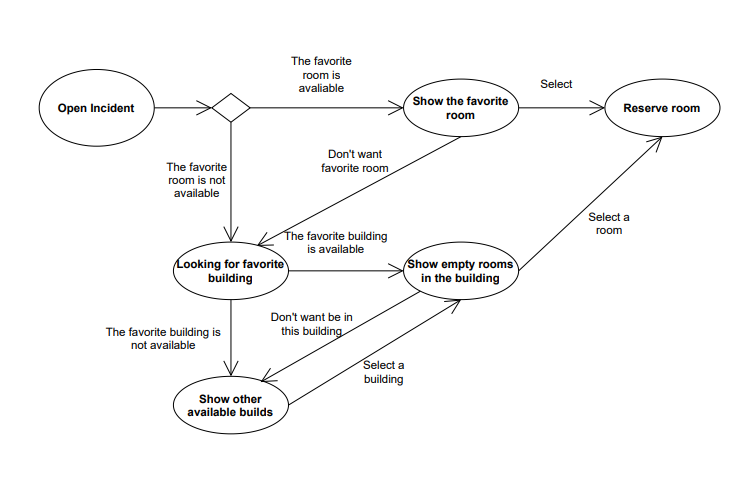


## Use Cases Diagram

The figure below illustrates the features that a user and an administrator have access to.

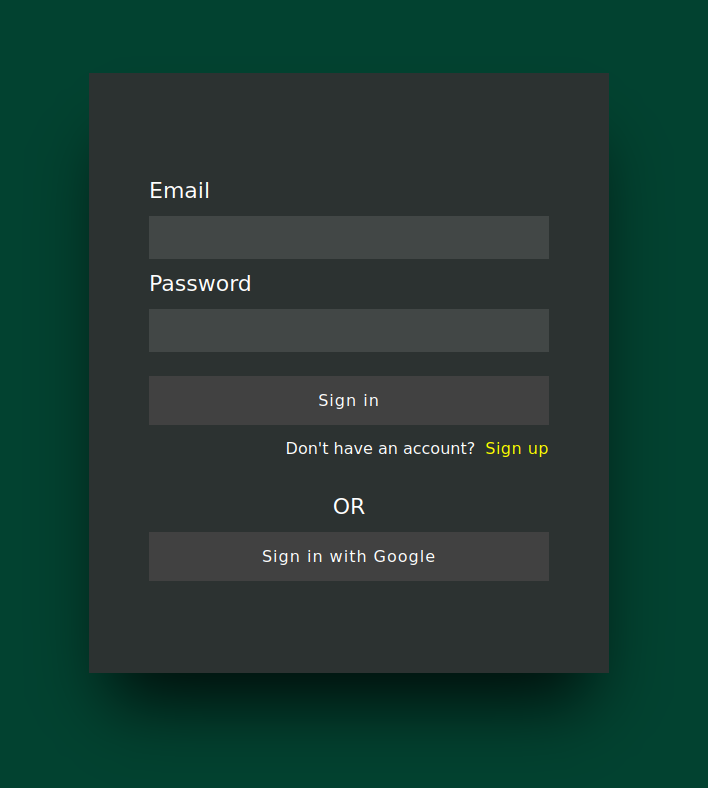


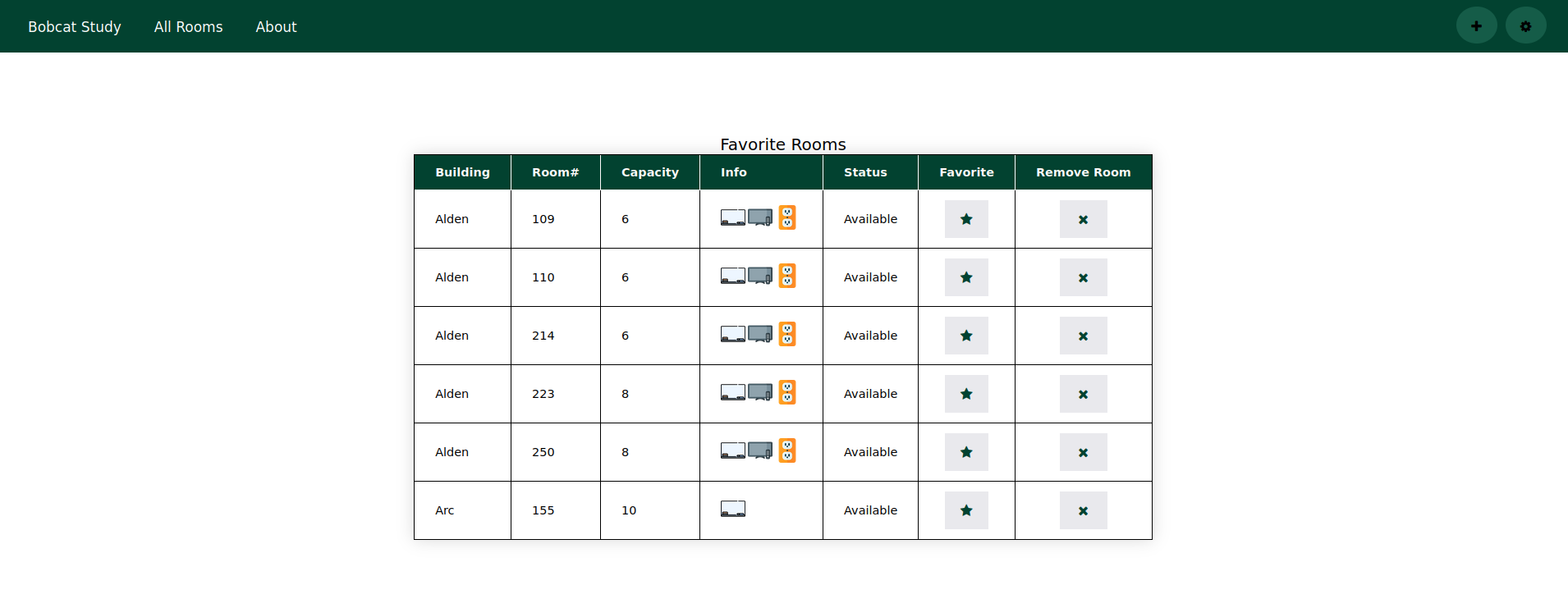
## Activity Diagram

The activity diagram shows steps to find and reserve a room.

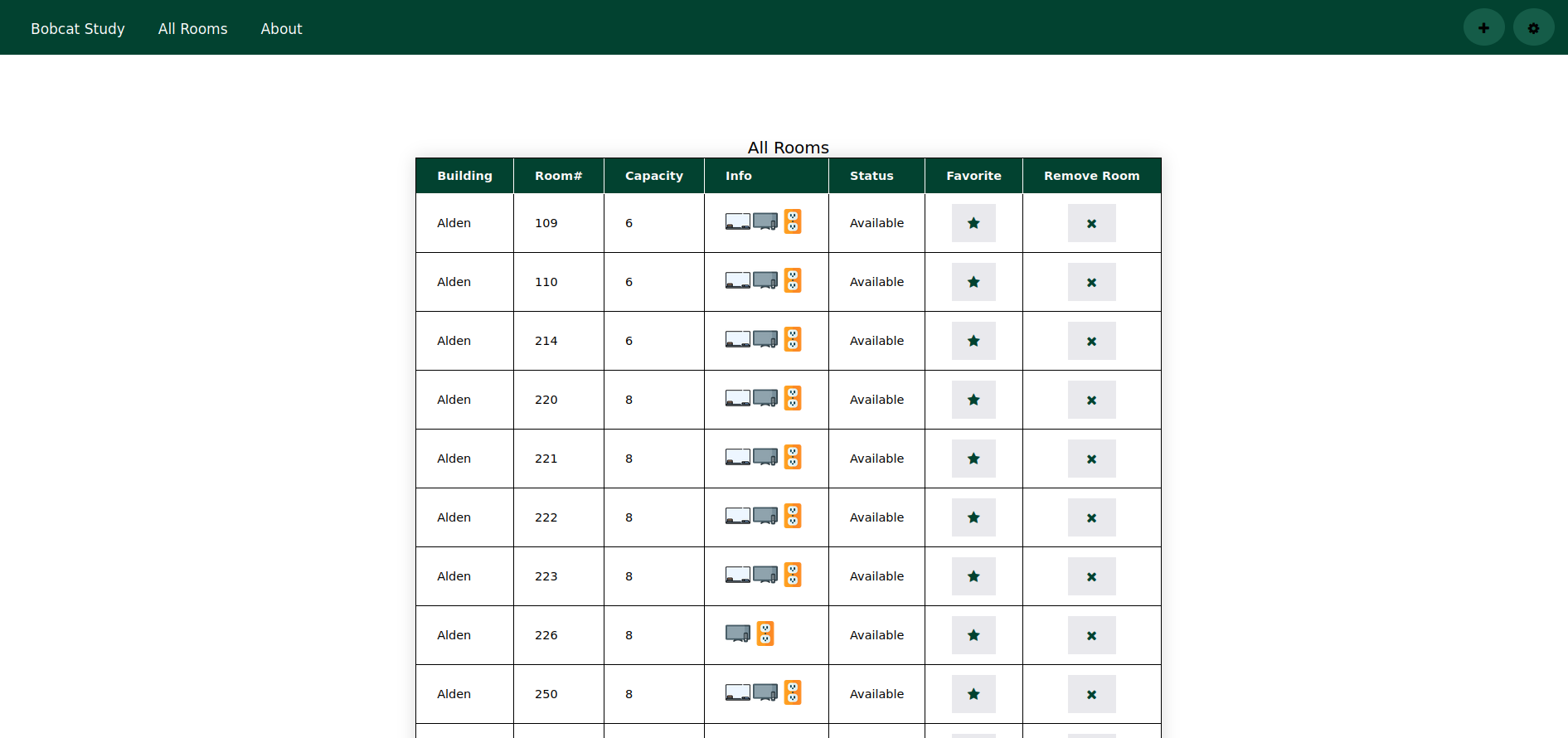
## Website Layout

This screenshot shows what our login / signup page looks like.



Our homepage will show the user’s favorites.

If the user has no favorites, they will be shown all the rooms.



There is also an about page with details about our project as well as all of our names and contact info.



## Technologies Used

For this project we are using React JS and React Native. The main reason behind choosing React was that our client requested that we use it. It is also a very popular framework used frequently throughout the industry so it will serve our project well. We chose it over other frameworks because we needed a framework that would allow us to develop for several different platforms and it was the framework our client requested that we use.

For account creation and authentication, we plan on using Firebase. Firebase is a widely used framework that allows for simple account creation and supports authentication through Google. This framework will allow us to have our users make accounts without worrying about the security risks of storing usernames and passwords on our own databases. It also allows users to circumvent the hassle of creating and remembering a new password just for our website by allowing them to choose to login with Google instead. We chose to use Firebase over other frameworks because it is the simplest to set up and the best way to add support for Google login to our project.

For our database we plan on using Firebase. We are already planning on using Firebase for authenticating our users through a Google account, so it will be very easy to add a database, as it is already integrated with our project. We chose to use Firebase over other services because we are using Firebase for our Google authentication which makes Firebase the best choice for databases as well by far.

For our webserver we plan on using Apache. Apache is free, open-source, and very widely used. We chose Apache over other services because we are already experienced with it and it supports several different platforms, so it will allow us to run it off of a Windows machine.