Where is the Peanut Room?

Just Janus

CIS 497 Fall 2018

Alec Austin ● Tim Bailey ● Kimberly Baum ● Matthew Larrimore ● Nikolas Parrent

Client: Mr. Howard Whitston

Table of Contents

Phase 1 Document

Project Proposal, Requirements, Stakeholders

Team Name: Just Janus ***Date:*** 9/5/18
Team Members: Tim Bailey, Kimberly Baum, Matthew Larrimore, Alec Austin, Nikolas Parrent
Project Name: Where is the Peanut Room? (WIP Title)
Project document storage location: GoogleDocs
Project coding repository: github

Project Client: Howard Whitston, University of South Alabama,
E-mail: ***hwhitston@southalabama.edu*** | *Phone: 460-7006* | *Fax: 460-7274*

1 Brief Project Description

An interactive map application which can provide directions between rooms in the building as well as information about the school of computing faculty.

2 Describe the problem your project will solve

Many individuals attending classes at Shelby Hall may have problems navigating the building if they are not familiar with their destination. This application will allow them to easily see what room they are looking for as well as some information about the person that is associated with the office/classroom.

3 *Stakeholders and stakeholder impact*

Stakeholder	***Impact of System***	***Explanation***
School of Computing Faculty	Low	Provides a means of easily finding other faculty and potentially new paths through the building

Shelby Hall Students and Visitors	High	Students and visitors unfamiliar with the building will be able more easily navigate to their intended destination, or to find information about various professors associated with certain rooms
CS, IS, IT, and HI Departments	Medium	Provides them with an easy way to give students directions through Shelby or to find information on a professor they ask about as well as a way of getting around for visitors they will have

4 Describe the Functional Requirements of the Project

- Users should be able to have a map that allows for listed directions from any room in Shelby to another room in Shelby with handicap accessibility.
- Users should be able to find out information about any faculty member in CS, IS, IT, and HI.
- Users should be able to find a list of all faculty members related to the topics of important groups (ACM, Day Zero, etc.), research opportunities, useful Freshman information (Vickie Speed, etc.), and labs.

5 Additional Desired Functional Requirements

- An optional guiding line on the map for directions.
- Additional information for harder to find places (such as the staircase behind the door on the first floor)

6 Nonfunctional Requirements

- System Requirements - Available on both IOS and Android platforms.
- Maintainability - Must be able to allow for changing room and instructor information
- Usability - Map and directions should be easy to understand and information on teachers should be easy to find

7 Technologies/skills expected and required

- MSSQL Server/ Server Management Studio
- C# OR High-Level App Development Software (TBD)

8 Currently known constraints

- Time
- Conflicting schedules
- Team/Client meeting availabilities

9 Use Cases

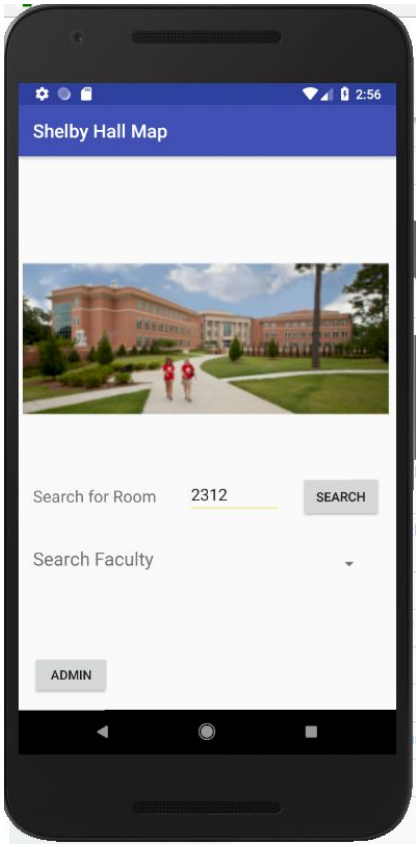
Use case:	Find a Classroom/Office

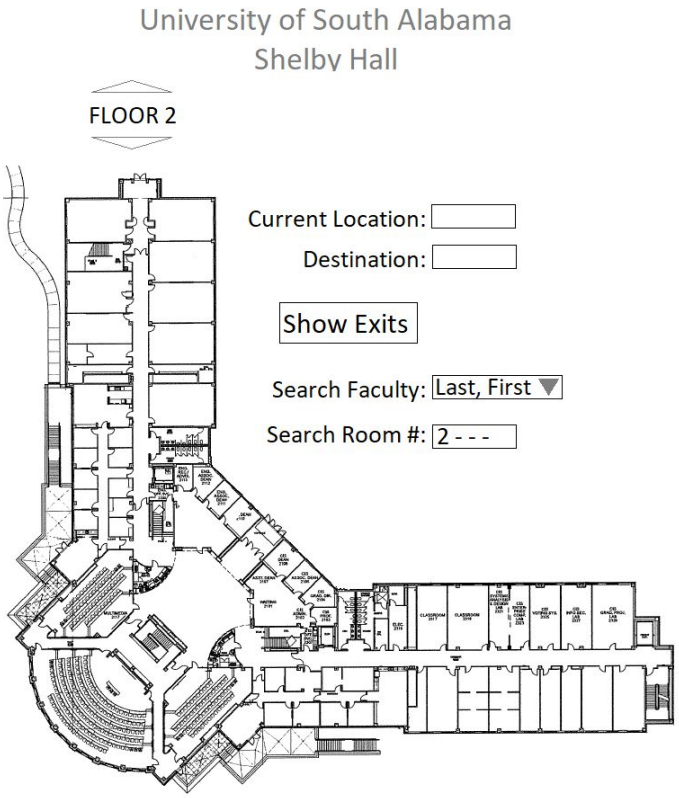
Actors:	Student
Preconditions:	Student must have the application installed. Searched room must be an existing room in Shelby Hall.
Postconditions:	The searched room is located on the map.
Flow of Events	
Actor Action	**System Response**
1. Student enters room number to find.	2. The system locates the room and shows it on the map.
3. Students indicate they want directions by inputting a room that they are near.	4. The system will give step by step directions
Alternative Flows of Events	
1. Student indicates that they are physically handicapped while requesting directions	2. System plans alternate route to use elevators to reach the destination

Use case:	Discover Information About Faculty
Actors:	Student
Preconditions:	Student must have the application installed. Searched faculty member/keyword must exist in database.
Postconditions:	The searched faculty member's information, or a list of existing things that the keyword could be referring to is shown.
Flow of Events	

Actor Action	System Response
1. Student enters faculty member's name to find out information about said faculty member.	2. The faculty member's information (such as classes taught, associated organizations, classrooms, and office hours) are displayed.
Alternative Flows of Events	
Actor Action	**System Response**
1. Student enters keyword (such as name, job title, organization, or office number) to find out information about the corresponding faculty member.	2. A list of corresponding faculty members is displayed.
3. Student selects, from the list, the faculty member that they are interested in viewing information about.	4. The faculty member's information (associated organizations and office number) are displayed.

10 System UI Prototypes





Phase 2 Document

1. **Detailed Use Cases**
   a. Use Case 1: Find a Classroom
   Actor: Student
   The student navigates to the map if he/she is not already there. The student then enters the number of the closest room to him/her and the number of the room to get to. The app then provides the student with directions from the student's current location to the desired location.
      i. Alternative Flow 1 - Outside of Building
      The student is outside of the building. In this case, the student will select the closest entrance to himself/herself.
      ii. Alternative Flow 2 - Accessibility Issues
      The student has some kind accessibility issue. The student will press the accessibility button, and the app will be provided adjusted directions that avoid stairs and other difficult places.

iii. Alternative Flow 3 - Tricky Path
The path to the destination goes through areas that are difficult to find. The app will provide special directions for locating these areas.

b. Use Case 2: Look up Faculty
Actor: Student
The student navigates to the faculty screen if he/she is not already there. The student types in the name of the faculty member he/she is trying to look up. The app returns the faculty member's information page.
   i. Alternative Flow 1 - Name Issue
   The student is unsure of the name of the faculty member. The student can instead have a list of all faculty members displayed in alphabetical order.
   ii. Alternative Flow 2 - Group Search
   The student is searching for a group of faculty instead of just one. The student chooses a group from the list of groups (faculty with a lab, faculty heading organizations, etc.) and will receive a list of all faculty associated with the chosen group.

c. Use Case 3: Emergency
Actor: Student
The student presses the emergency button. The student is immediately shown a map with all emergency exits labeled and is given directions to the nearest one.
   i. Alternate Flow - Accessibility Issue
   The student presses the accessibility button and is given directions that avoid difficult places for people with accessibility issues.

d. Use Case 4: Useful Information
Actor: Student
The student navigates to the useful information page. The page displays information about important faculty members and locations in Shelby Hall.

2 System Requirements

We plan to follow an agile development methodology to confirm that we meet all of the following requirements as well as adjust for any changes to requirements that may occur as the project progresses.

2.1 Functional Requirements

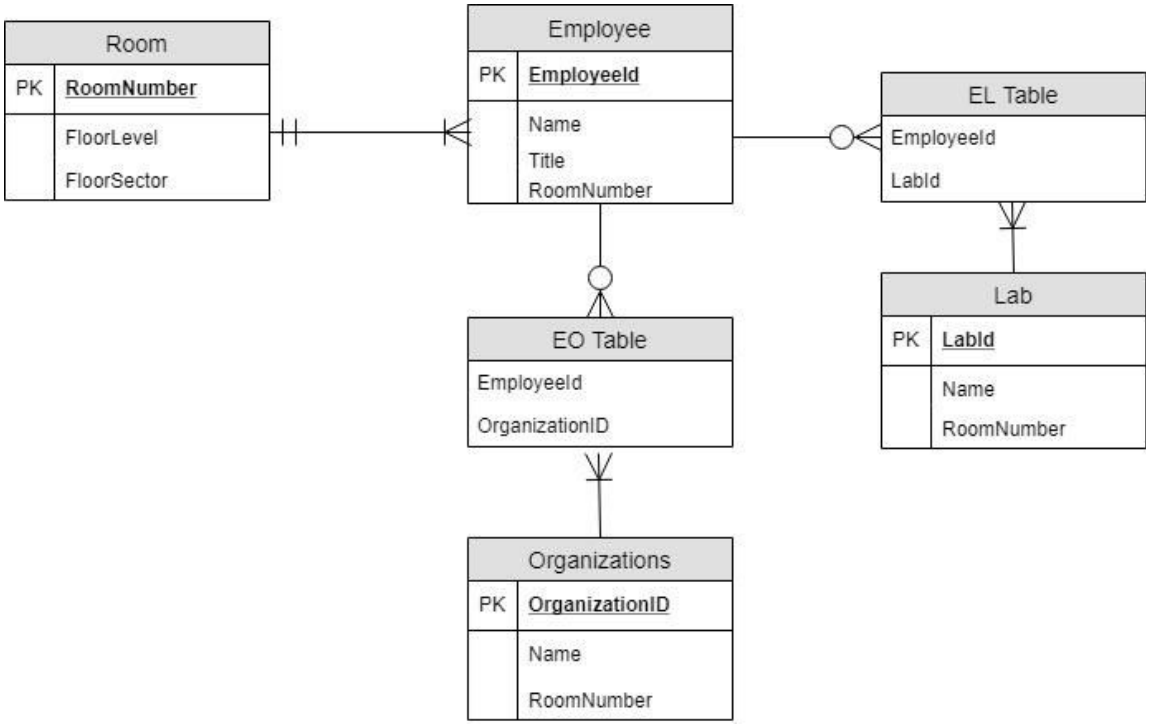
a. The first functional requirement is for our application to have a map with listed directions for moving through Shelby Hall. This will be handled by having a scrollable, zoomable image of the Shelby Hall map on a map screen. Then, that map will be divided into 5 sections, which will be used to determine what set of directions will be displayed in a box below the map for the user to read. We will also handle our fifth functional requirement of having special directions for hard to find places here, be making sure any directions through hard to find areas are very specific and provide extra information to find the correct way.

b. The second functional requirement for our application is for anyone using the application the be able to find information on any faculty member in the CS, IS, IT, and HI departments. This will be handled by having a page where a user can enter the name of a faculty member and be given a page to look at based on that faculty member. This page will have information pertaining to the faculty member that the faculty member can enter him/herself through a login. Faculty information will be stored in a SQL Server Database.
c. The third functional requirement is to have useful information for freshmen. This will be accomplished by having a page of our app dedicated to useful information that will be easily accessible from our navigation page.
d. The fourth functional requirement is optional and is to have a guiding line. If we have time to do this, this will be handled as an overlay drawn on the map.

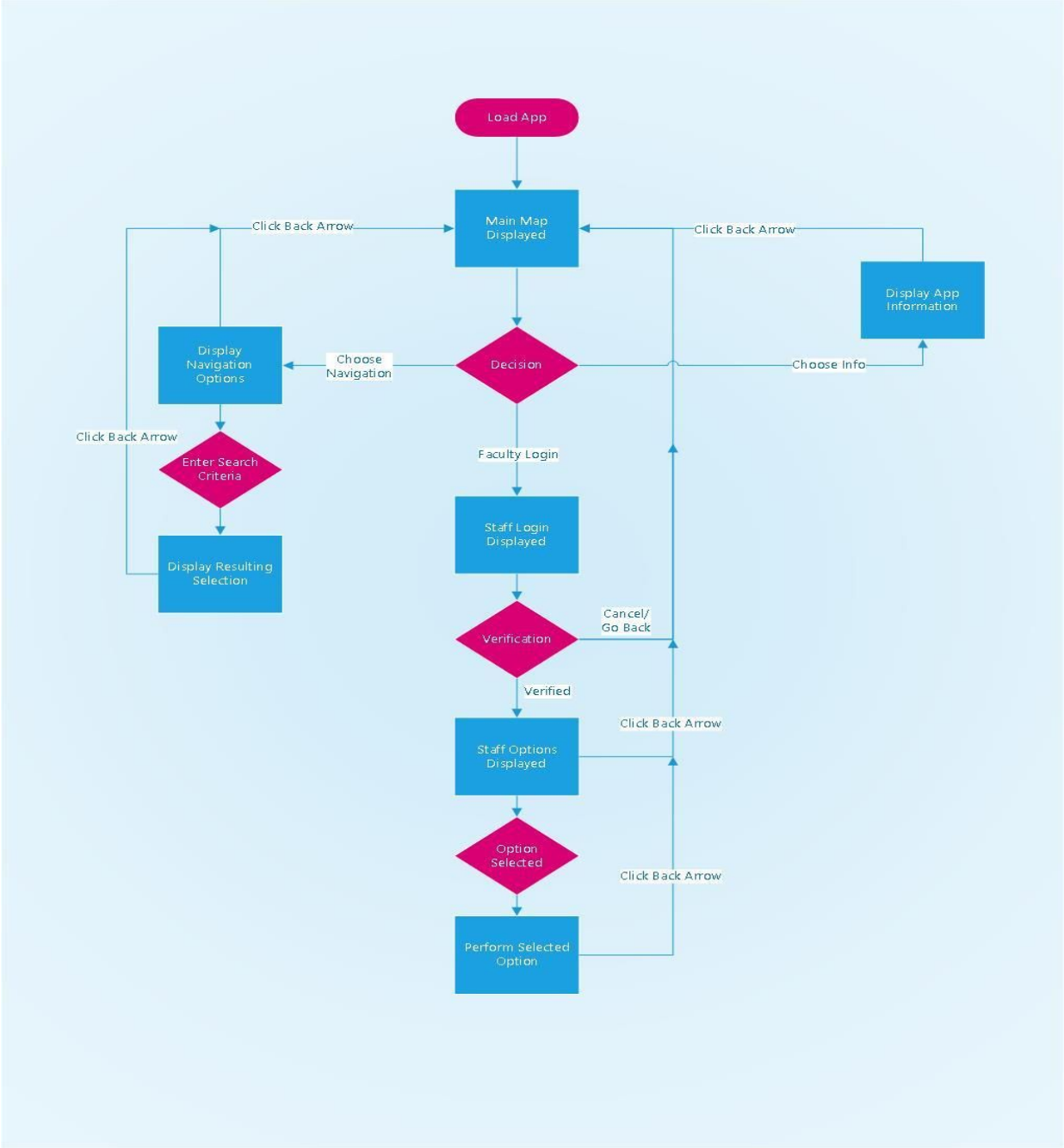
2.2 Nonfunctional Requirements

a. Our first nonfunctional requirement is to have the app on Android and iOS. We plan to implement our system requirements of iOS and Android by using Flutter. Flutter is a framework that allows for simultaneous iOS and Android programming using its language called Dart.
b. Our second nonfunctional requirement is to make our app easy to maintain. This will be done by having a well-organized database as well as instructor capability to change their information pages to keep everything up to date.
c. Our third nonfunctional requirement is for our app to be easy to use. This will be done by having a very clear and sensible navigation page, a map image that is clear and easy to read, a faculty page that is simple to search through, and directions that are not difficult to follow.

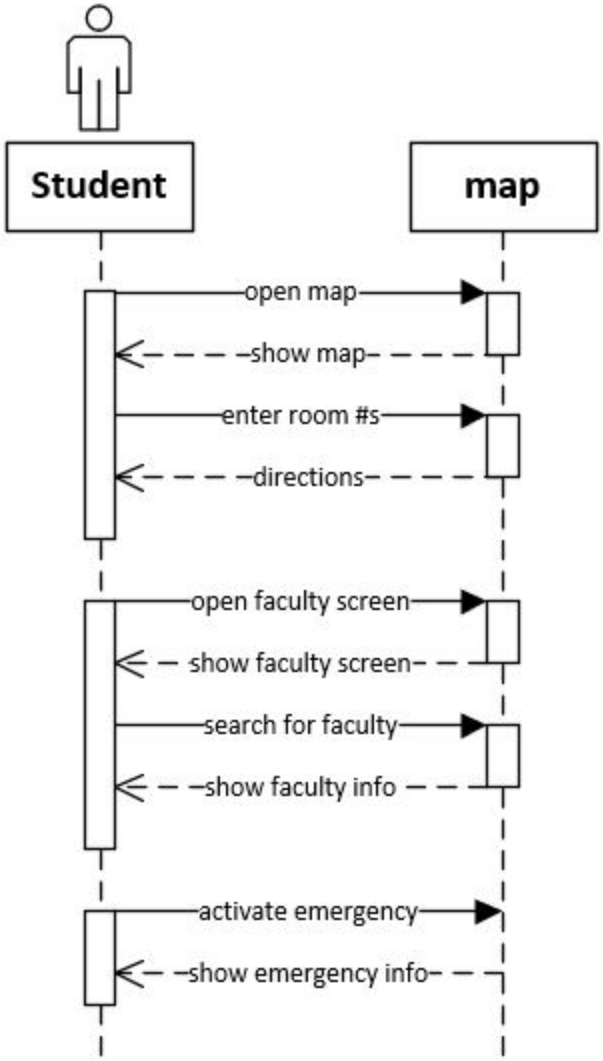
3 ERD


Room
PK
RoomNumber
FloorLevel
FloorSector
Employee
PK
EmployeeId
Name
Title
RoomNumber
EL Table
EmployeeId
LabId
Lab
PK
LabId
Name
RoomNumber
EO Table
EmployeeId
OrganizationID
Organizations
PK
OrganizationID
Name
RoomNumber

4 Flow Chart


Load App
Main Map Displayed
Click Back Arrow
Click Back Arrow
Display App Information
Display Navigation Options
Choose Navigation
Decision
Choose Info
Click Back Arrow
Enter Search Criteria
Faculty Login
Staff Login Displayed
Display Resulting Selection
Verification
Cancel/ Go Back
Verified
Click Back Arrow
Staff Options Displayed
Option Selected
Click Back Arrow
Perform Selected Option

4 Sequence Diagram



Phase 3 Document

1 Introduction

1.1 Project Background

The Shelby Hall Map application will allow students to easily see what room they are looking for as well as search for faculty in Shelby Hall School of Computing.

1.2 Objectives

The test plan is to define tests so we can ensure that all requirements are met.

1.3 Testing Strategy

The Shelby Hall Map application will be tested manually by the developers during all development phases.

2 Test Items

2.1 User Interface

The user interface will be tested manually by all team members as defined by the test cases.

2.2 User Manual

The user manual will be a part of the Phase 3 deliverables. The manual will be assessed by all team members to ensure the manual is clear and helpful to new users.

2.3 Operator Procedures

Just Janus will ensure that there is a user manual and administrative instructions for users. The Shelby Hall Map application will work according to requirements in the development and test environments. Access to all resources will be provided to Mr. Whitston for future development and maintenance.

3 Features to test

- Administrator Login
- Search for room
- Open map
- Directions
- Faculty information pages
- Home page loading

4 Untested Features

There aren't any untested features to list at this time.

5 Test Approach

5.1 Component Testing

Each developer is responsible for testing the component that they are assigned to develop. Components will be tested during each phase of development against the component test cases.

5.2 Integration Testing

All components will be tested to ensure they work according to the requirements. Integration test cases will define the proper conditions that must be met for proper integration of the components.

5.3 Conversion Testing

There isn't new data that will be used for conversion, so conversion testing isn't necessary.

5.4 Production Environment Testing

Production environment testing will occur at the end of development. The current development environment emulates the required hardware.

5.5 Interface Testing

Interface testing will ensure the buttons are clickable, test is easy to read and the user interface flow is friendly. Tests will be conducted to ensure the proper data is provided on user requests.

5.6 Security Testing

The administrator login page will be tested to ensure correct behavior. Only users with the administrator username and password should be able to log in as administrator.

The system should provide input validation. SQL Injection attacks and other bad inputs should not be allowed.

5.7 Recovery Testing

Recovery testing is not planned at this time.

5.8 Performance Testing

Performance testing will be a performed during all test cases. During testing, it should be noticed if the application is slower than expected.

5.9 Regression Testing

Any part of the system that has been changed must have their associated test cases performed. This will verify the current working parts of the application still work under the new changes.

5.10 Acceptance Testing

The Shelby Hall Map will be testing against all requirements to ensure the application works as planned. This final phase of testing will make sure the application is correct according to the predefined requirements.

5.11 Beta Testing

Shelby Hall Map will not be beta tested.

6 Pass/ Fail Criteria

6.1 Suspension Criteria

If the application is not responsive or fails during the execution of a test case, the testing phase will be halted until the application is running under normal conditions again.

6.2 Resumption Criteria

Testing will continue when the application is in a working state. Testing will only occur when there are active test cases to verify.

6.3 Approval Criteria

If the application works as intended during the execution of a test case. The test case will be given a “pass” and saved in the test log.

7 Testing Process

7.1 Test Deliverables

Test deliverables will include:
Test Log (every test performed)

Test #		Area/Component	
Pass/Fail	*Pass/Fail*		
Tested by		Date	
Criteria			
Comments			

Test log template
Issue Log (all issues found during development and testing)

Issue #		Area/Component	
State	*Active/Resolved*	Severity	*Low/Medium/High*
Entered by		Date	
Description			
Repro Steps			
Verified by		Date	

Issue log template

7.2 Testing Tasks and Responsibilities

- Design and implement test cases for good and bad input.
- Log test case results in the Test Log.
- Log issues in the Issue Log.
- Verify fixed issues.

7.3 Schedule

Task	**Deliverable**	**Week**	**Due Date**
Develop test cases	Test Cases	10-12	November 7, 2016
Run and log tests	Test Log	10-15	December 6, 2018
Log issues	Issue Log	10-15	December 6, 2018
Project Showcase	PowerPoint Presentation Shelby Hall Application	15	December 6, 2018

8 Environmental Requirements

8.1 Hardware

- Microsoft Windows PC

8.2 Software

- Android Studio 3.2.1 for Windows 64-bit
  https://developer.android.com/studio/
- Java SE Development Kit

https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html

- Flutter SDK
  https://flutter.io/setup-windows/
- Flutter Plugin
- Dart Plugin
- GitHub access to Shmapp repository
- Google drive to view Test and Issue logs

8.3 Tools

All tests will be executed manually, so testing tools will not be used.

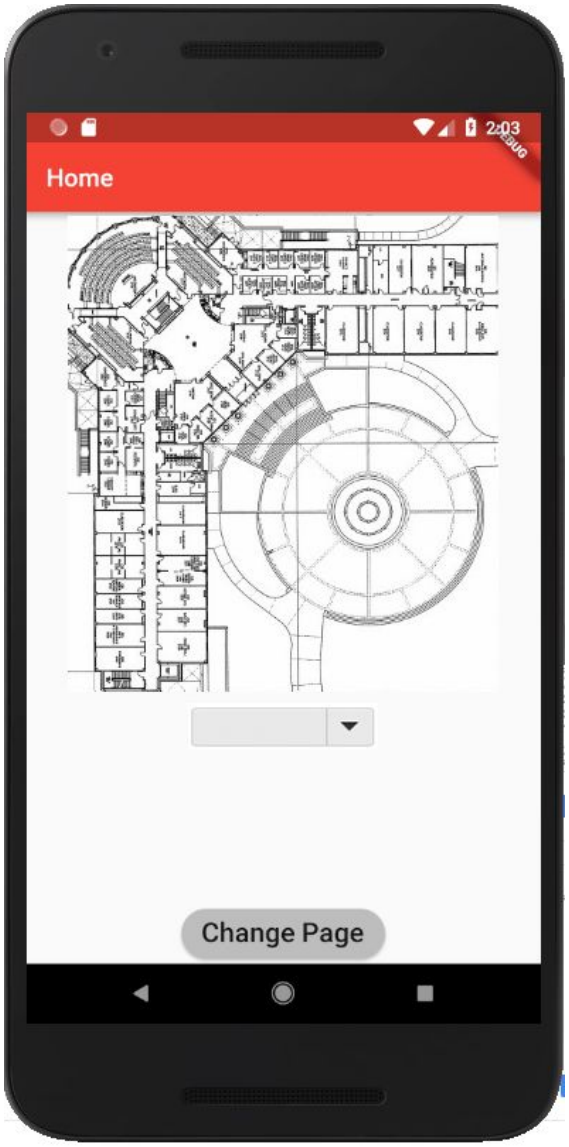
8.4 Risks and Assumptions

- The Shelby Hall application will be verified using the chosen development environment. The application may not run on all hardware.
- The Shelby Hall application will include an administrative login. If a student guesses the login, they will have administrator access to the application.
- Students may not be willing to download the application to their mobile device.

User Manual

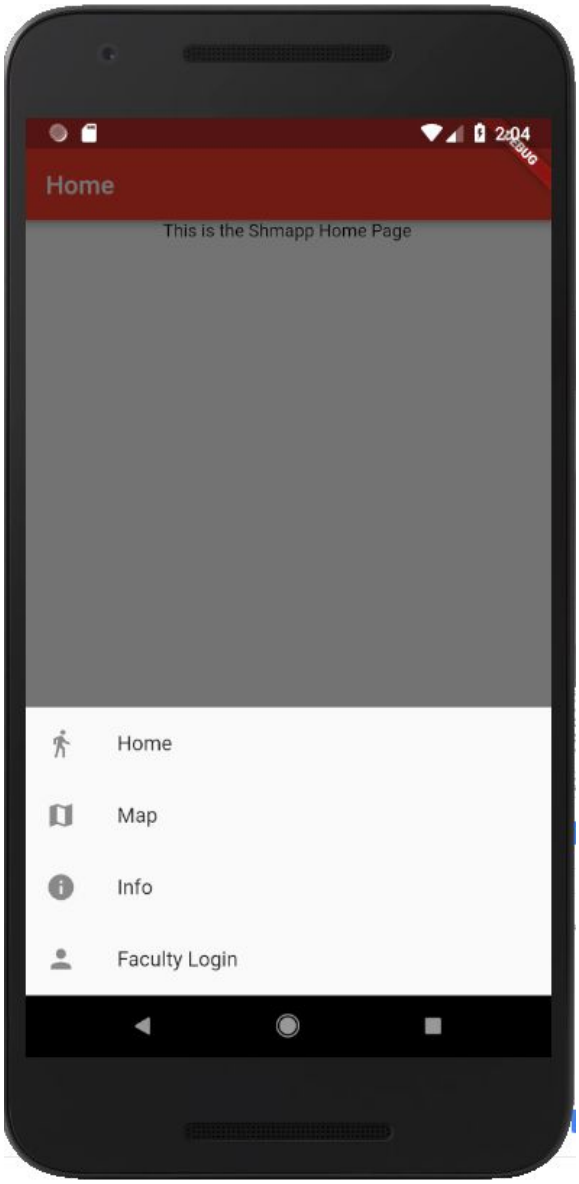
Contents:

Main Screen



When opening the application, you will be greeted with a resizable map of Shelby Hall. You are able to view any floors of the building by using the dropdown menu directly below it

Page Navigation

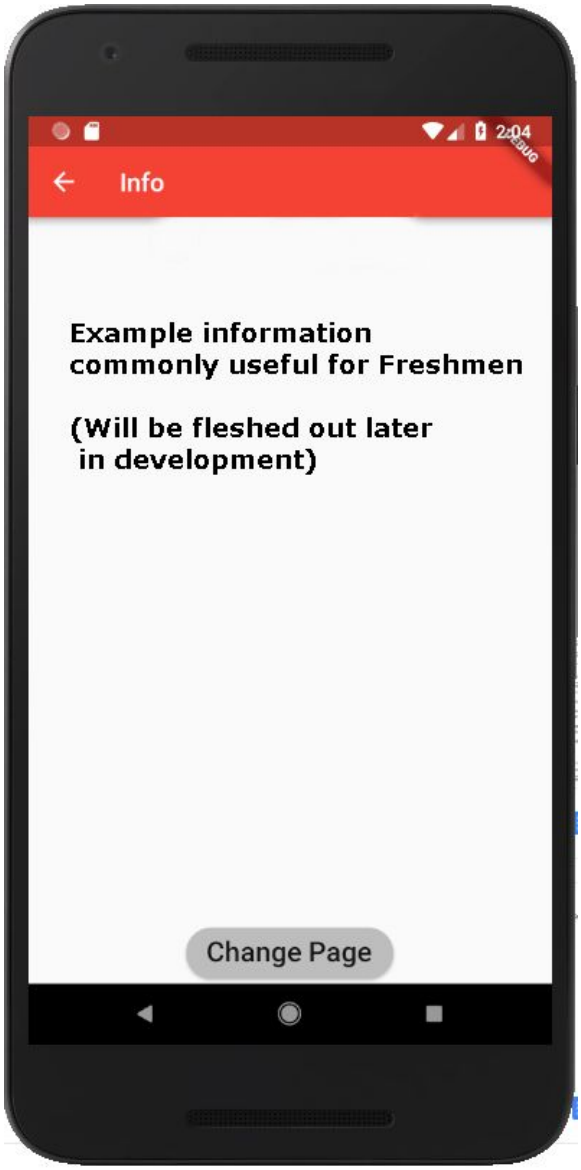


To navigate to the other pages available in this app, click the “Change Page” button at the bottom of the screen. A small menu at the bottom will appear. From here, you can choose which page
you wish to view.

Map

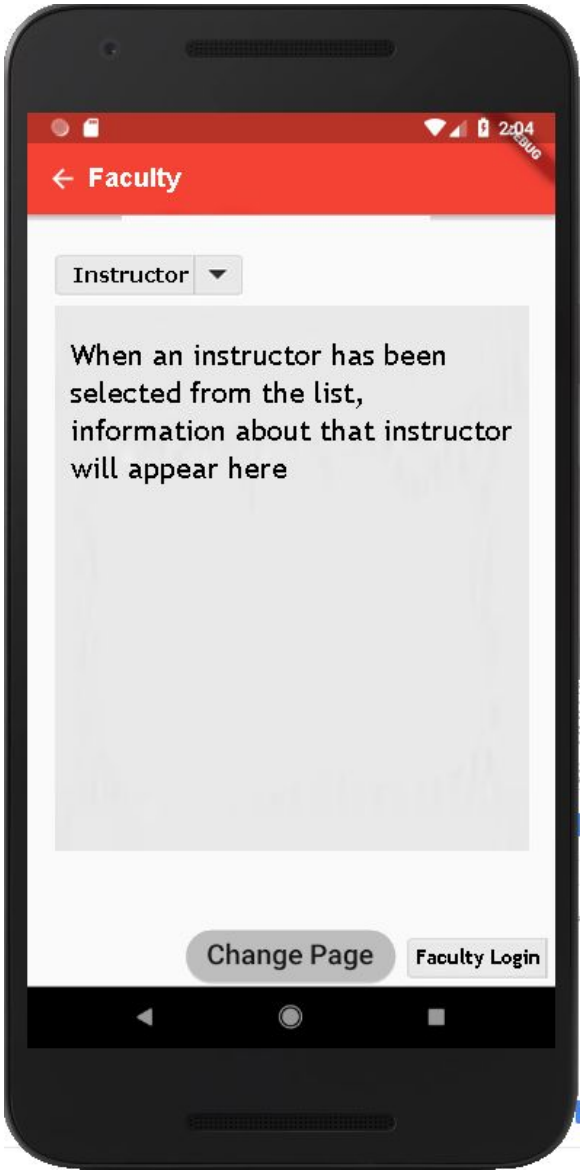
When viewing the Map page, you will be presented with the same map as the Home page, but you now have the option to input where you are and where you want to be. The app will display your directions when you have confirmed your Start and End locations.

Info



On the Info page, Information that is regularly used to help Freshmen will be displayed in a FAQ fashion. You will be able to scroll through the information that are encased in several dropdown hiders depending on the information you wish to view.

Faculty



On the Faculty page, a textbox is available for you to search several faculty of Shelby Hall. It also auto fills with several potential matches and can be used as a dropdown for easier searching. When a faculty member is selected, relevant information regarding them will be displayed below.

For faculty members who wish to alter their information, a Faculty Login button is provided on the bottom right. Clicking it brings you to the Faculty Login screen.

Faculty Login

Upon clicking the Faculty Login button on the previous screen, you will be prompted with a login screen. Using the login credentials given to you (if you are a faculty member of Shelby Hall), you will be able to alter the relevant information that presents itself when your name is selected.

Phase 4 Document

1 Assessment of Functional Requirements

1.1 Functional Requirements

- Users should be able to have a map that allows for listed directions from any room in Shelby to another room in Shelby with handicap accessibility.
  - This was successfully completed
- Users should be able to find out information about any faculty member in CS, IS, IT, and HI.
  - This was successfully completed
- Users should be able to find a list of all faculty members related to the topics of important groups (ACM, Day Zero, etc.), research opportunities, useful Freshman information (Vickie Speed, etc.), and labs.
  - Groups and freshman opportunities are completed. Labs were completed as well as we could find information on who had them. Research opportunities were not completed because we could not find the relevant information in time.

1.2 Additional Desired Functional Requirements

- An optional guiding line on the map for directions.
  - This was not successfully completed as it proved itself too difficult to complete with our limited time.
- Additional information for harder to find places (such as the staircase behind the door on the first floor)
  - This was mostly completed. There are a few rooms that are harder to find that do not have special directions because of time, but most of the hard to find places do have extra instructions to find them.

3 Test results

12/2/2018	Alec Austin	Opening the Application	Performance Test	Pass

Criteria	Opening the Application takes less than 3 seconds
Results	Successfully opens in less than 3 seconds

12/5/2018	Kim Baum	Home/Change Page	Performance Test	Pass
Criteria	Change page from Home takes less than 2 seconds			
Results	Changing from Home to Info took less than a second			

12/5/2018	Kim Baum	Home/Select floor	Performance Test	Fail
Criteria	Select floor takes less than 2 seconds			
Results	Select floor is not functional.			

12/5/2018	Kim Baum	Home/Select floor	Performance Test	Pass
Criteria	Swipe map takes less than 2 seconds			
Results	Swipe map takes less than 2 seconds.			

12/5/2018	Kim Baum	Map/Change Page	Performance Test	Pass
Criteria	Change page from Map takes less than 2 seconds			
Results	Changing page from map takes less than a second.			

12/5/2018	Kim Baum	Map/Get Directions	Performance Test	Pass
Criteria	Get directions takes less than 2 seconds			
Results	Get directions takes less than 2 seconds			

12/5/2018	Kim Baum	Map/Go to previous page	Performance Test	Pass
Criteria	Going to previous page from Map takes less than 2 seconds			
Results	Going to previous page from Map takes less than a second.			

12/5/2018	Kim Baum	Info/Change page	Performance Test	Pass
Criteria	Change page from Info takes less than 2 seconds			
Results	Changing page from info takes less than 2 seconds			

12/5/2018	Kim Baum	Faculty/Change page	Performance Test	Pass
Criteria	Change page from Faculty takes less than 2 seconds			
Results	Changing page from Faculty takes less than 2 seconds.			

12/5/2018	Kim Baum	Faculty/Search Faculty name	Performance Test	Pass
Criteria	Search for a faculty name takes less than 2 seconds			
Results	Searching for a faculty member takes less than a second.			

12/5/2018	Kim Baum	Faculty/Search organization	Performance Test	Pass
Criteria	Search for an organization takes less than 2 seconds			
Results	Searching for an organization takes less than a second.			

12/5/2018	Kim Baum	Home/Change Page	Feature Test	Pass
Criteria	Change page from Home brings user to selected screen			
Results	Change page bring user to the correct screen.			

12/5/2018	Kim Baum	Home/Select floor	Feature Test	Fail
Criteria	Selecting floor on the home page shows the selected floor			
Results	Select floor is not functional.			

12/5/2018	Kim Baum	Map/Change Page	Feature Test	Pass
Criteria	Change page from Map brings user to selected screen			
Results	Change page bring user to the correct screen.			

12/5/2018	Kim Baum	Map/Get Directions	Feature Test	Pass
Criteria	Get directions returns directions			
Results	Entering 2 existing rooms and selecting get directions returns directions.			

12/5/2018	Kim Baum	Map/Go to previous page	Feature Test	Pass
Criteria	Selecting back arrow from Map goes to the correct previous page			
Results	Selecting back arrow from Map brings user to Home page.			

12/5/2018	Kim Baum	Info/Change page	Feature Test	Pass
Criteria	Change page from Info brings user to selected screen			
Results	Change page bring user to the correct screen.			

12/5/2018	Kim Baum	Faculty/Change page	Feature Test	Pass
Criteria	Change page from Faculty brings user to selected screen			
Results	Change page bring user to the correct screen.			

12/5/2018	Kim Baum	Faculty/Search Faculty name	Feature Test	Pass
Criteria	Search faculty name returns correct faculty			
Results	Correct faculty shown.			

12/5/2018	Kim Baum	Faculty/Search organization	Feature Test	Pass
Criteria	Search organization returns correct organization.			
Results	Selecting back arrow from Map brings user to Home page.			

12/5/2018	Kim Baum	Home/Change Page button	Interface Test	Pass
Criteria	Change page button on Home works correctly and position and size are correct			
Results	Change page button on Home is correct.			

12/5/2018	Kim Baum	Home/Select floor button	Interface Test	Fail
Criteria	Select floor button works correctly and position and size are correct			
Results	Select floor is not functional.			

12/5/2018	Kim Baum	Home/Map size	Interface Test	Pass
Criteria	The map on the home page is readable and zoomable			
Results	The map is readable, zoomable and can be swiped.			

12/5/2018	Kim Baum	Map/Change Page button	Interface Test	Pass
Criteria	Change page button on Map works correctly and position and size are correct			
Results	Change page button on Map is correct.			

12/5/2018	Kim Baum	Map/Start input	Interface Test	Pass
Criteria	Start input box editable and readable			
Results	Start input box is editable and readable			

12/5/2018	Kim Baum y	Map/End input	Interface Test	Pass
Criteria	End input box editable and readable			
Results	End input box is editable and readable			

12/5/2018	Tested by	Map/Accessibility button	Interface Test	Pass
Criteria	Accessibility button works, and correct information is given			
Results	Accessibility button works, and correct information is given			

12/5/2018	Tested by	Map/Get Directions button	Interface Test	Pass
Criteria	Get Directions button works, and correct information is given			
Results	Get Directions button works, and correct information is given			

12/5/2018	Kim Baum	Map/Direction text readable and correct	Interface Test	Pass
Criteria	Direction text readable and correct			
Results	Direction text readable and correct			

12/5/2018	Kim Baum	Map/Previous Step button	Interface Test	Pass
Criteria	Previous step button works, and correct information is given			
Results	Previous step button works, and correct information is given			

12/5/2018	Kim Baum	Map/Next Step button	Interface Test	Pass
Criteria	Next step button works, and correct information is given			
Results	Next step button works, and correct information is given			

12/5/2018	Kim Baum	Map/Map Size and sizing	Interface Test	Pass
Criteria	The map on the home page is readable and zoomable			
Results	The map on the home page is readable and zoomable and can be swiped to change map.			

12/5/2018	Kim Baum	Map/Text size and readability	Interface Test	Pass

Criteria	Text on the Map page is readable and correct
Results	Text is readable after zooming in.

12/5/2018	Kim Baum	Info/Change Page button	Interface Test	Pass
Criteria	Change page button on Info works correctly and position and size are correct			
Results	Change page button on Info is correct.			

12/5/2018	Kim Baum	Info/Text is readable and correct	Interface Test	Pass
Criteria	Text on the Info page is readable and correct			
Results	Text is readable and correct			

12/5/2018	Kim Baum	Faculty/Change Page button	Interface Test	Pass
Criteria	Change page button on Faculty works correctly and position and size are correct			
Results	Change page button on Faculty is correct.			

12/5/2018	Kim Baum	Faculty/Search input	Interface Test	Pass
Criteria	Search input is entered correctly with autocomplete			
Results	Autocomplete works			

4 Lessons learned section

Interface design is a tricky business. Looking back now, taking everything that is displayed in the GUI and having it as the child of a Container object would have improved the interface greatly and allowed for more control over the design and spacing.

We originally attempted to use text messages and GroupMe to message each other. We quickly realized that we would need a platform that would allow us to have conference meetings while away from the Wednesday and Thursday class sessions due to the number of conflicting schedules. We settled on Discord because it was free, light to download and use, allowed us to be able to host a conference call, and supported screen-sharing. We could have used Skype for Business, but most of us were already familiar with the Discord platform.

We were originally using MSSQL Server but discovered that it does not play well with flutter or dart. As such we had to switch databases mid project which cost us a little bit of time in database implementation.

Working with new tools is always difficult. We had many problems with sharing code that would work in all of our environments.

5 Future Direction section

A future senior project team could add a guiding line to the map, create a login feature to allow instructors to change their information on the faculty page, add research opportunities to the presented information on either the info page or the pages of faculty members related to the opportunities, improve the clarity or directions and add additional directions for harder to find

rooms, change the database to be hosted on a remote server and accessed through an API, or eventually extend the structure of this app to other buildings on campus (such as HAHN.)

6 Technical Hand-off section

6.1 **MySQL Database:**

For our project we set up a MySQL database containing tables corresponding to School of Computing faculty members, organizations, Shelby Hall rooms, and labs along with intersection tables between them. In its current state, the database is hosted on the localhost per individual machine and directly connects to and queries the database via a dart library called SQLJocky5. Given more time we would set up the database on a remote server and access the database via an API for both functionality and security purposes.

6.2 GitHub:

We have a github repository set up consisting of a main branch and a sub-branch. The main branch contains the main.dart file that handles a majority of all of the features of the project while the sub-branch contains the same main.dart file but adds the MySQL connectivity and implementation. We did this so that members of our team could work on separate features without compromising the other members' branch in any way.

Source code can be located at: https://github.com/KotoKuraken/shmapp