

D o c u m e n t a t i o n

Campus Maps

Enjoy Campus, One Step at a Time

JEFF WILSON
JUSTIN VINEYARD
MELISSA CLAUDE
ALBERT VARGAS
LUIS LEON

Table of Contents

| | |
|------------------------------------|-----------|
| Overview | 3 |
| Requirements | 4 |
| User Requirements | 4 |
| System Requirements | 5 |
| Functional | 5 |
| Non Functional | 6 |
| Architecture | 7 |
| Client-Server Model | 7 |
| Generic-Layered Architecture Model | 8 |
| Implementation | 9 |
| Use Case Diagram | 9 |
| State Diagram | 10 |
| Sequence Diagram | 10 |
| Testing | 11 |
| Plan Overview | 11 |
| Units | 11 |
| Format | 11 |
| Feedback System | 12 |
| Overview | 12 |
| Testing Plan (Chronological Order) | 13 |
| Script (Pseudocode) | 14 |
| Sequence Diagram | 15 |
| Destination System | 16 |
| Overview | 16 |
| Testing Plan (Chronological Order) | 16 |
| Diagram | 17 |
| Pseudocode | 17 |
| System Capacity | 18 |
| Overview | 18 |
| Testing Plan (Chronological Order) | 18 |
| Pseudocode | 19 |
| Conclusion | 20 |

Overview

Campus Maps is a product designed by students, for students. This application will aid students in getting the most out of their college experience by providing them up to date information about their campus. Available resources will be made more easily accessible to its users. Additionally, major events taking place on campus will be reported via the application allowing students to better understand the events taking place at their school. In this product, the students have the voice. Registered students will be able to publish their favorite routes among the campus, helping others. Students can create events on the app that correspond to activities taking place currently such as protest, demonstrations, voting stations, and many others! By polling the feedback of other students, the event can be used to create delays in their calculated transit time and suggest better routes. The students are in control of their campus experience with this application!

Requirements

1. User Requirements

- 1.1. The Campus Map system shall verify the user is registered.
- 1.2. The system shall allow users to select their current campus.
- 1.3. The Campus Map system shall display a map of the selected campus that displays levels of traffic among the campus.
- 1.4. The system should allow the user to select a departure and arrival point that will be used to calculate the estimated time in transit and arrival time for their current route.
- 1.5. The system shall allow the user to publish and view “Favorite Routes”
 - 1.5.1. The system shall allow users to publish their route for other users to view
 - 1.5.1.1. The system shall allow users to draw their route on the map.
 - 1.5.1.2. The system shall ask the user for the departure and arrival points of the route.
 - 1.5.1.3. The system shall ask users for their estimated time in transit.
 - 1.5.1.4. The system shall ask users for any notes they wish to include with the route overview.
 - 1.5.2. The system shall display an overview for each published route among campus.
 - 1.5.2.1. A highlight of the published route shall be shown on the map
 - 1.5.2.2. The estimated time in transit of the route shall be displayed to the viewer
 - 1.5.2.3. Any notes the publisher created when publishing the route shall be displayed to the user.
- 1.6. The system shall allow the user to create points of expected delay along the campus.
 - 1.6.1. The system shall verify the integrity of the delays through the feedback of other users.
 - 1.6.2. The system shall display a brief description of what is causing the delay.
 - 1.6.3. The system shall notify other users of the delay .
- 1.7. The system shall show bus stops on the campus and be interactive with the user.
 - 1.7.1. The system shall allow the user to select a bus stop to display additional info about it.
 - 1.7.1.1. The system shall display what bus routes currently serve this stop.
 - 1.7.1.2. The system shall display how busy the stop currently is.

- 1.7.1.3. The system shall display the estimated time between bus arrivals.

2. System Requirements

2.1. Functional

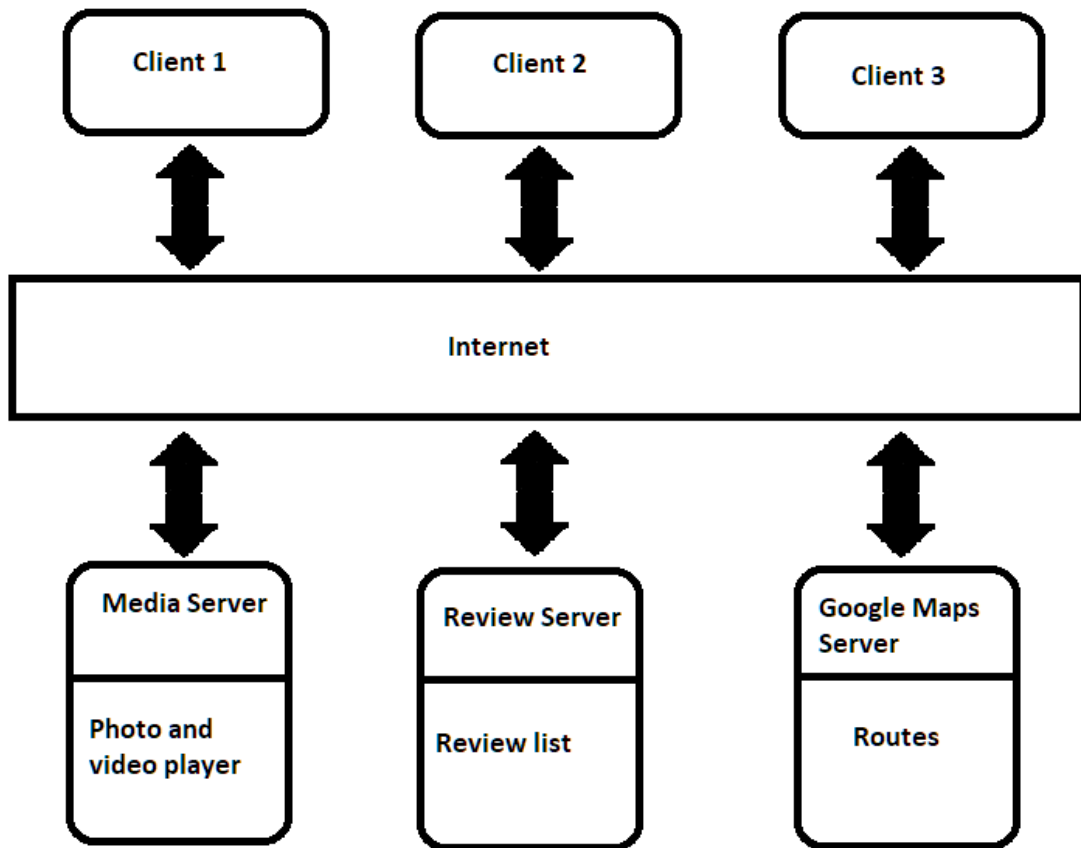
- 2.1.1. System shall use a login screen to verify the user is registered.
 - 2.1.1.1. The login screen shall prompt for the Username and Password of the user.
 - 2.1.1.2. The users should be given 5 attempts to login before they are locked out for 30 minutes.
- 2.1.2. System shall display a Top-Down Two Dimensional Perspective map of the campus
 - 2.1.2.1. Popular walkways shall be drawn on the map
 - 2.1.2.2. Populated walkways shall be shown in an orange hue that gets darker and shifts to a red the busier it becomes
- 2.1.3. System shall collect route data from users once the route is finished
 - 2.1.3.1. System should store data for one hour
 - 2.1.3.2. System shall collect time in transit from user
- 2.1.4. System should utilize crowd sourced data submitted from users to calculate the fastest possible route
 - 2.1.4.1. System shall reference user submitted delays
- 2.1.5. System should use user traffic data (location history) to create trends of quickest routes.
- 2.1.6. System should have users verify populated events
 - 2.1.6.1. System shall display a Green Checkmark (Confirm) and Red X (Deny) next to event so users can validate its existence
 - 2.1.6.2. System shall remove an event if at any point there are 5 more “Deny” submissions than “Confirm” submissions on the event.
- 2.1.7. System shall ping wifi connections/GPS signals to determine more populated areas
- 2.1.8. System should update routes every 10 minutes
- 2.1.9. System should notify user of important events and changes
 - 2.1.9.1. If a change in route time is detected, the system shall notify the user of the new estimated time of arrival
 - 2.1.9.2. If system fails to update, the system shall notify the user that “Route Update Unsuccessful, routes may not reflect current conditions”

2.2. Non Functional

- 2.2.1. System shall adhere to all privacy policies for the device platform
- 2.2.2. System shall have access to necessary resources on the user's device
 - 2.2.2.1. Global Positioning System (GPS)
 - 2.2.2.2. Wireless Internet (WIFI) and Mobile Data
- 2.2.3. System shall allow up to 1500 users to register
- 2.2.4. System shall support 1000 of active users
- 2.2.5. System shall have access to the Google Maps API
- 2.2.6. System shall be ADA compliant
 - 2.2.6.1. System shall recommend routes based on the required accommodations for user
- 2.2.7. Utilize system notification services (will vary per platform)
- 2.2.8. Needs to target lowest possible software version to increase user base
- 2.2.9. Shall be available at all hours of the day

Architecture

Client-Server Model



Generic-Layered Architecture Model

3-Dimensional Perspective Map User Interface

Campus Maps Application

Configuration services

Database Management

Group Management

Application services

Real Time Traffic Updates

Resource Finder

User-Generated Updates

User-Published Routes

Map for University

User-Driven Feedback

Utility services

Identity Verification

Google Maps Interfacing

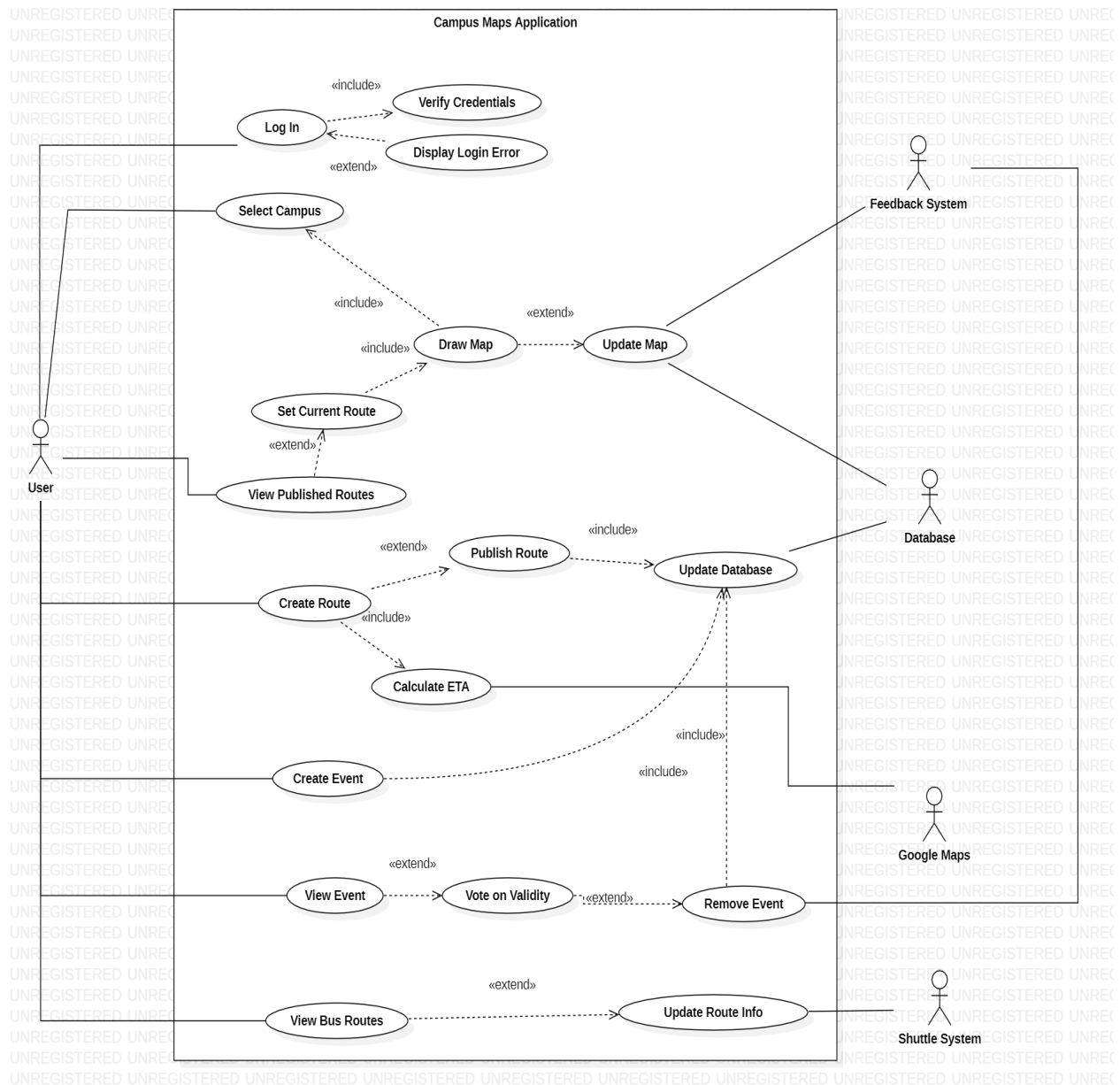
Shuttle Service Interfacing

GPS Monitoring

Application Storage

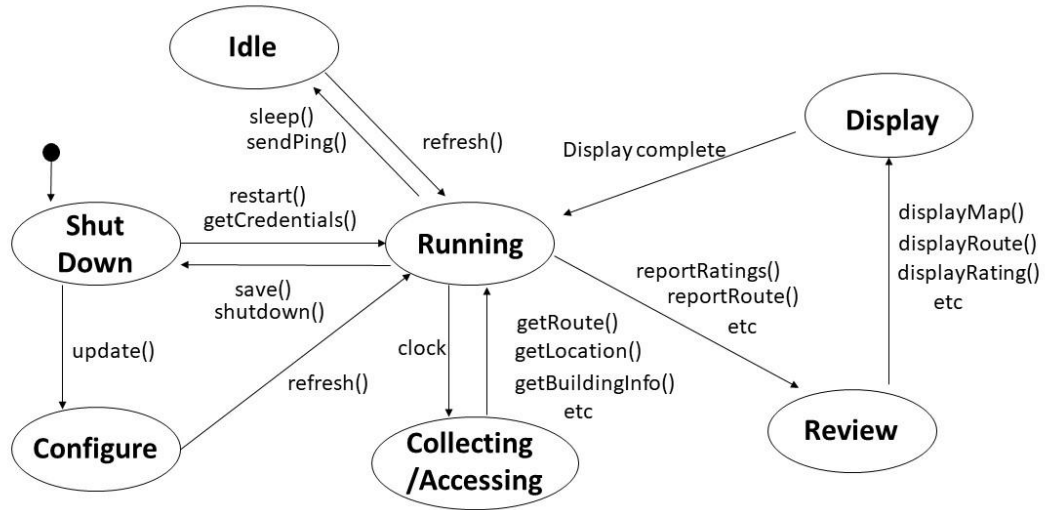
Implementation

Use Case Diagram



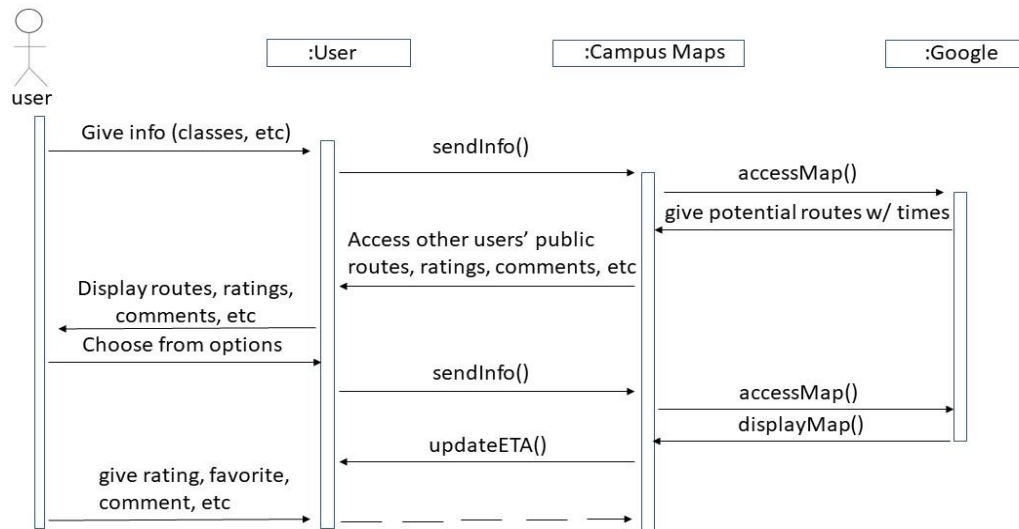
State Diagram

Campus Maps Condensed State Diagram



Sequence Diagram

Campus Maps Simple Sequence Diagram – getting a route



Testing

Plan Overview

This Testing Plan outlines several critical systems and how they will be tested, with the majority of the testing being unit testing.

Units

This plan tests the functionality of each of these units:

- Feedback System
 - Ensuring that the user can vote on the validity of an event, and that the event is suggested for removal upon certain conditions.
- Destinations
 - Ensure that the user is routed correctly and efficiently.
- System Capacity
 - Can the Campus Maps systems hold the amount of active/registered users, and how it handles too many users attempting to log on.

Format

Each Chronological Testing Plan will be written in the following format:

1. What event/action is trying to be validated (or system to be purposefully broken).
 - a. What conditions are necessary, and what we will introduce to test the validity of the event.
 - i. What we expect to happen if the task is implemented correctly and the test is run successfully.

Feedback System

Overview

The Feedback system is an integral part of a user-moderated application. This system allows users to verify the existence and/or importance of a user-created event. If the threshold of negative votes on an event is reached, the event should be removed from every user's map. This system will also communicate with the main database of events, it does not store any events. However, it acts as a filter to the database system by recommending which events be removed. The Feedback will only hold the ID numbers of the existing events located in the database, as well as the amount of suggestions for removal each event has. In order for this system to work as intended it must be connected to the database system.

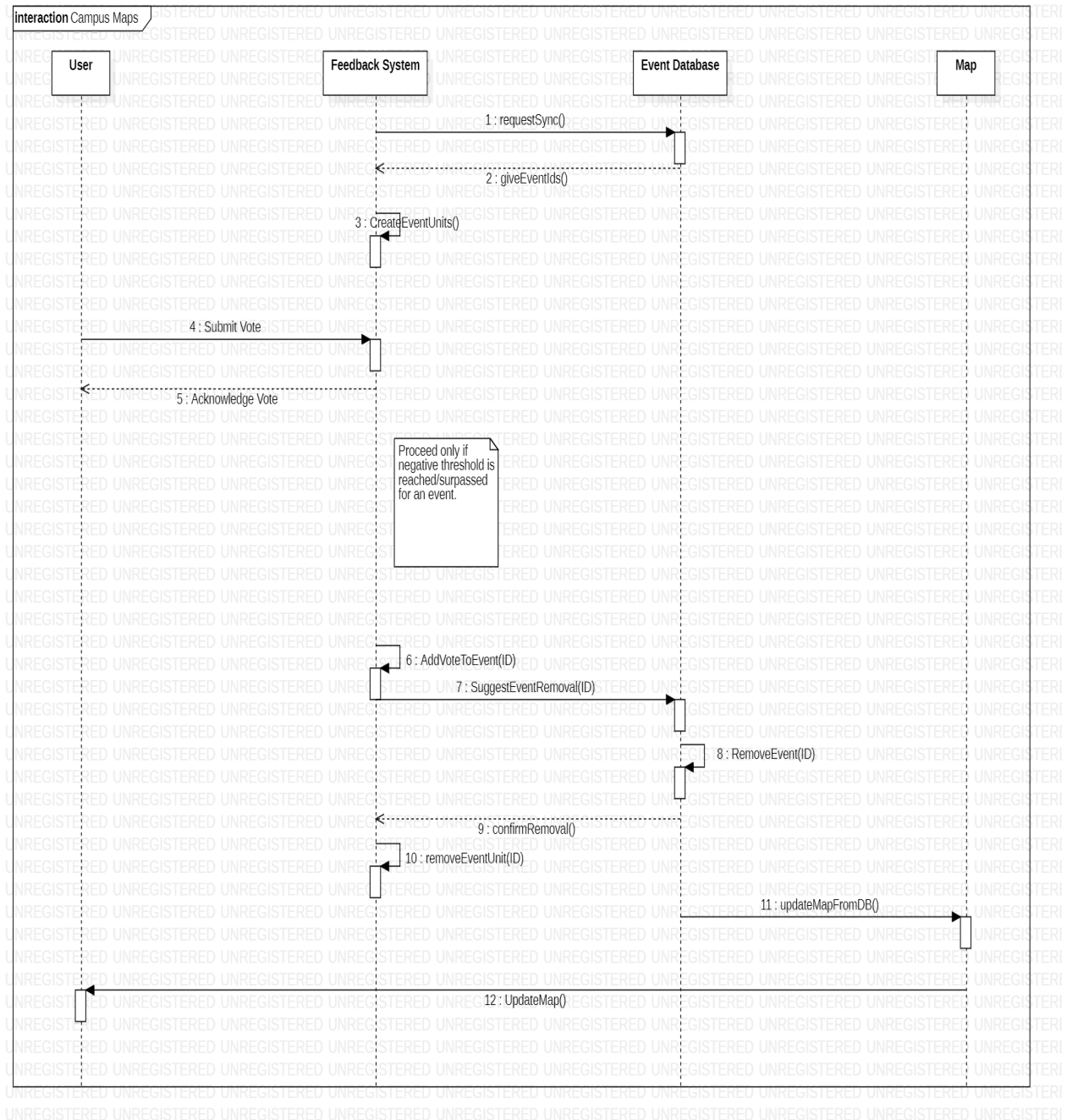
Testing Plan (Chronological Order)

1. Ensure the Feedback System can communicate with the Database System.
 - a. With existing events on the database, have the feedback system request from the database system regarding the ID of every existing event.
 - i. If this task is implemented correctly, the Database and the Feedback System should have a matching list of event ID numbers.
2. Ensure the Feedback System is kept up to date with the Database of events.
 - a. With a matching list of Event ID numbers on both the Database and the Feedback systems, remove an event from the database.
 - i. If this task is implemented correctly, the feedback system should receive a ping from the database system that the events have changed. The feedback system will then sync up with the database and have a matching list of event ID numbers.
3. Ensure the feedback system accurately keeps track of how many negative votes an event has.
 - a. With an existing event and a matching list of events between the Feedback System and the Database, vote on the validity of events (either positive or negative votes) from multiple devices.
 - i. If this task is implemented correctly, the list of events in the Feedback System should have a corresponding amount of negative votes to the respective event ID number.
4. Ensure the Feedback system recommends that the database remove an event once the negative response threshold has been met or surpassed.
 - a. With an existing event and the feedback system in sync with the database, vote on the event with 5 negative reviews from different devices.
 - i. If this task is implemented correctly, the Feedback System should send a notification to the database calling for the removal of an event, and then should request to sync with the database system.

Script (Pseudocode)

- Create an event on a client device (Device A)
- Locate Event on another client device (Device B)
 - Verify B can see event from A
- On Device B and 4 Other Client Devices, give a negative vote on the event from each device.
 - Event should now be removed
- Vote on event from Device B
 - Application should inform Device B and its user that the event no longer exists and will be removed from MAP.

Sequence Diagram



Destination System

Overview

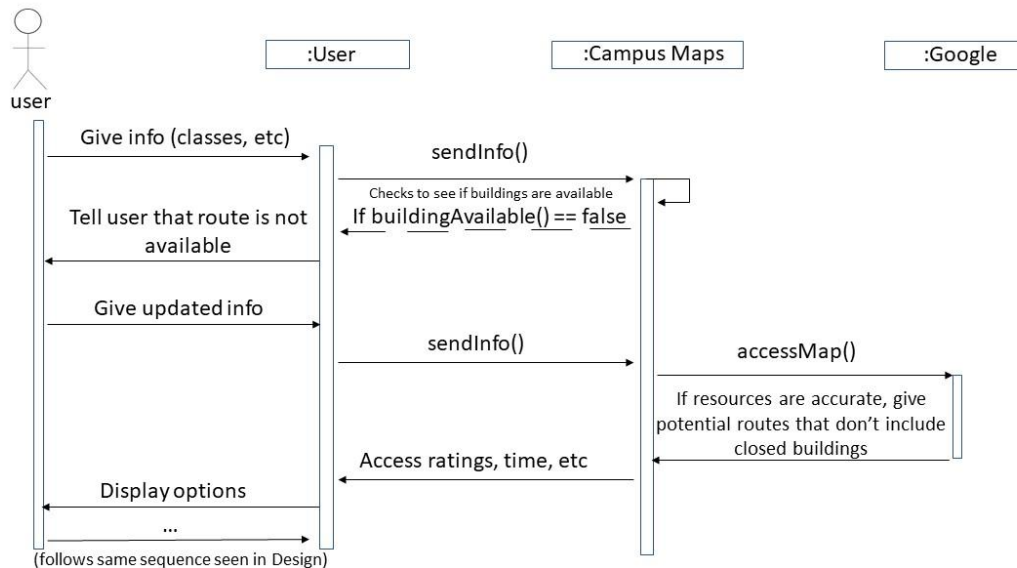
One of the advantages of Campus Maps compared to other destination applications is the ability to utilize buildings to allow the route to be more efficient. During working hours, users shall have the option to utilize the building to get to their destination. The destination application shall give routes that have the user going through a campus building. However, once the building closes, it shall not allow the user nor give the option to go through the building. This test will ensure that the application only offers routes that can be accessible after hours. There are **two** tests happening with the destination system: i) the user is allowed to request a route during any time of the day and ii) the user will be denied a route if a building is closed when it's after hours.

Testing Plan (Chronological Order)

1. Ensure that Google Maps works with Campus Maps during normal operating hours to give routes that utilize all campus buildings
 - a. When the user requests for a route anywhere on campus, Google Maps shall give the user a variety of routes
 - i. If implemented correctly, the routes offered will include routes that utilize campus buildings, along with routes that are accessible, top rated, etc.
2. Ensure that Google Maps works with Campus Maps after hours
 - a. When the user requests for a route anywhere on campus, Google Maps shall give the user a variety of routes that are available during this time frame
 - i. If implemented correctly, users shall still have the ability to find routes to and from different locations that are allowable
3. Ensure that Google Maps works with Campus Maps and does not offer routes that include buildings that are closed
 - a. When the user requests for a route anywhere on campus that includes buildings, Google Maps shall not give the user route options that utilize buildings that are closed
 - i. If implemented correctly, users shall still have the ability to have access to routes that do not utilize other buildings, but gets routes that go around buildings
 - ii. A pop up should appear telling the user that the route they requested is unavailable during that time, if it is a saved route, and offer the next efficient route

Diagram

Campus Maps Simple Sequence Diagram – requesting a route (after hours)



Pseudocode

while getRoute()

 if **normalHours()** == true

 EXPECTED RESULT:

- User receives routes that allow them to use buildings to get to places
- User receives routes that don't use buildings
- User is allowed to input buildings for their route
- User will be allowed to give and receive ratings
- ...

 else

 EXPECTED RESULT:

- User receives a notification that buildings are closed and cannot be utilized
- User will not receive routes that utilize buildings
- User will receive routes that don't utilize buildings
- User will not be allowed to input buildings for their route
- User will be allowed to give and receive ratings
- ...

System Capacity

Overview

Having a system capacity set is important when it comes to maximizing the performance of the application along with delivering the best user experience. The systems shall allow up to 1500 users to be registered and support 1000 active users. However, once any of these limits are reached the system shall notify the user that the capacity has been reached. These tests will ensure that the system continues to function while at capacity and that the system does not allow registration or log in once capacity is reached.

Testing Plan (Chronological Order)

1. Ensure the system works with up to 1500 registered users
 - a. With less than 1500 registered users, attempt to register with a new account
 - i. If implemented correctly, the system shall go through the registration process without any issues and a new account shall be created
 - b. With 1500 registered users, attempt to register with a new account
 - i. If implemented correctly, the system shall not allow the user to register and shall give an error message letting the user know that it cannot register at the moment.
2. Ensure the system works with up to 1000 active users
 - a. With less than 1000 active users, attempt to log in and perform basic actions on the application
 - i. If implemented correctly, the system shall allow the user to log in and use the application without any issues
 - b. With 1000 active users, attempt to log in and perform basic actions on the application
 - i. If implemented correctly, the system shall not allow the user to log in and notify the user that the server is currently at capacity and suggest trying again later

Pseudocode

While user is registering

Get amount of registered users

if(registered users > capacity)

Expected Result:

- User shall not be allowed user to register
- User should receive a message indicating that it's unable to register

else

Expected Result:

- User shall be registered into the system
- User should receive an email indicating that it was registered
- User should see visual confirmation that registration was successful

While user attempts to log in

Get amount of active users

if(active users > capacity)

Expected Result:

- User shall not be allowed to log in
- User should receive a message indicating that the application is currently at capacity and to try again later

else

Expected Result:

- User should be able to log in
- User should be able to see all his settings set on a previous session
- User should be able to view campus map
- User should be able to use the application's core features

Conclusion

Campus Maps is more than just a destination application. It's an application that promotes efficiency, safety, and community. With the ability to find the best route to class (in terms of time, scenery, or accessibility) to being aware of what is causing buzz around campus (protests, demonstrations, free items), Campus Maps will take care of it all. Students recommendations and feedback will help the community thrive to help with potential threats or benefits. This application allows each individual to use their voice and take back control of the ever-changing college experience.

Campus Maps is first introduced with the **requirements** section. Here, you can see all the information that this application has to offer. **Architecture** dives deeper and shows the “insides” of how it all will work. With the **design**, a few examples of how certain user actions are demonstrated. Finally, **testing** illustrates tying this all together to make sure Campus Maps works to the best of its ability!

However, the capabilities of Campus Maps doesn't stop there! In future versions, users will have the ability to check to see if there is parking available, the average wait time for dining services, and checking vehicle traffic around the surrounding areas.

College is stressful enough. Incoming freshmen, super seniors, faculty, and staff will enjoy the use of this application alike. Students shouldn't be stressed about getting to class and they should be aware of their options. It's time for students to get the most out of their college experience and be in the know of what is happening a few steps away. Campus Maps helps make the college experience one step easier!