

"System and Unit Test Report"

GeoBus(iOS)
Bus Squad
11/30/2015

System Test Scenarios for Sprint 1

User Story 1: As a user, I want to be able to open the app and immediately see a Google map of UCSC showing a single campus shuttle on it so that I know how close it is to my current location.

Scenario:

1. Start GeoBus app
2. Check to see if Google maps is opened properly
3. User should see the bus marker updating and moving around the map.
4. Check at a bus stop to make sure the bus location is accurate.

User Story 2: As developers, we want to set up a git repository so that we can all collaborate on the source code and track the evolution of the project.

Scenario:

1. Open your web browser and go to github.com
2. Make an account if you don't have one already.
3. Make sure the rest of the developers have an account and make a group repository so everyone can collaborate and work with each others code.

System Test Scenarios for Sprint 2

User Story 1: As a user, I want to be able to see all active shuttles updating in real time in order to determine which bus best suits my destination on campus.

Scenario:

1. Start GeoBus app
2. Make sure Google maps is up and running and that it doesn't crash under no circumstance
3. User should see all shuttle buses that are currently active so he can determine which bus is best suited for his ride to another destination.

User Story 2: As developers, we want to have started writing our documentation so that the project can still have a pulse once we finish CMPS 115.

Scenario:

1. Open up google docs, or any other document sharing website
2. Share the document by providing the email for all participants.
3. Write useful information about the code and how you figured out some problems so others know what you were doing or what your code does.

System Test Scenarios for Sprint 3

User Story 1: As a user, I would like to know the schedules of any Metro buses just in case I live off-campus so that I can plan my ride home accordingly.

Scenario:

1. Open GeoBus app
2. See the button that allows us to see the different bus schedules
3. Choose the bus that you want to see the schedule for
4. Close the window and go back to the map with the buses updating

User Story 2: As a user, I would like the app to be as simple as possible so I won't have to spend time learning how to use the app.

Scenario:

1. Start GeoBus app
2. Make sure the sidebar works so I can easily access the Metro bus schedule
3. If I close the sidebar, make sure the app still has the Google maps running
4. Close the app to finish the testing

User Story 3: As developers, we want to finalize the project's documentation so that any future teams that work on it will have a clear description of how we accomplished everything.

Scenario:

1. Keep adding documentation to the shared file
2. Also add product features that we wanted to implement but had trouble doing so and explain what troubles we had when we tried to implement it
3. Have your team also contribute to this document
4. Put the documentation on the GitHub account so everyone can have access to this document.

Unit Testing:

The testing that I performed was to make sure the XML file would update properly and the buses would show up in the correct position. I would open up the app and sit at different bus stops to make sure that the raspberry pi that is stationed at each bus stop functions properly. Every bus stop seemed to work perfectly and it would properly display the bus at the correct position on the app. The only bus stop with an issue is the bus stop in front of OPERs gym. I went to try to figure out the problem and realized the raspberry pi at the bus stop needs to be programmed again as a power outage caused the device to fail and once in awhile causes the buses to freeze and jump across the map. This problem is currently being worked on but the problem now is less persistent.

I conducted a testing to find any hidden bug/ unforeseen interaction inside the app. Basically I load up the application and start messing with all the bus stops, bus icons, and the map itself to see if something would cause the app to crash. For example, I tried to select two buses simultaneously. Same with the bus stops. Also tried to keep the bus that is out of service selected. Nothing seems to be out of place and everything seems to be functional.

A lot of the problems we had was when we were merging different parts of our code together as it would bring up problems that weren't previously there. Another big problem we ran into was that the map would constantly refresh, thus clearing everything that was on the map but that was fixed just by only updating the latitude and longitude of the buses and keeping everything else unchanged which really made the app run faster as well.

Another problem that occurred even before the testing phase was when the IDE for Xcode was updated, Xcode was trying to force us out of using Google Maps and using Apple Maps but since we were deep in the Sprint we had to figure out how to downgrade Xcode so that we could still use Google Map's API. Also, a lot of features were scrapped due to the problems they were creating like mapping the bus route using polylines as it wouldn't come out cleanly but it would be really nice to implement this feature in the future.

Known Issues:

1. Bus will once in awhile jump around randomly (rarely)
2. App sometimes crash when switching data providers
3. Bus stop markers text box doesn't work properly