**Deliverable 3**

Anja Sheppard

Group 11

11/19/18

CS 1200.011

Ethan Yong, Timothy An, Reg Gonzalez

Develop a mobile attendance system: This product is an app that students download onto their phone. It tracks students’ whereabouts during class time to see if they are actually attending class.

My sub-problem: GPS Tracking

1. User interface prototype

There is no user interface for this portion of the project. The GPS tracking is all back-end code, as it is used to determine whether students are attending class. No interface is needed because student location is private to other users. The interface is integrated into the app in other ways, such as in the reports section where students can see their attendance track record.

1. Pseudocode

There are a lot of member functions of classes not described here, such as adding a class or getter/setter functions. These are self-explanatory in their own right so I decided to describe the super juicy and important functions, like updating the GPS and syncing with Galaxy.

updateGPS(): void

* retrieve GPS coordinates of the student’s phone, update longitude and latitude vars
* if the GPS location is not attainable (due to being inside)
  + retrieve location via the UTD CometNet WiFi Access Points
* change lastChecked to the current time/day/month/year in a string

syncWithGalaxy(): void (more of a high-level description that pseudocode)

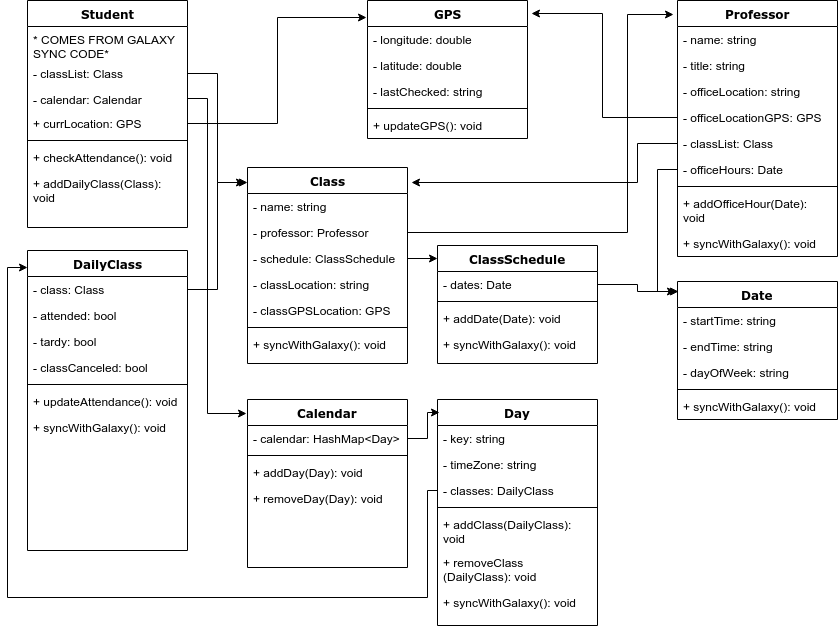
* There are many of these functions spread throughout the classes
* They are all run at 2 hour intervals from the main function
* They each interact with Galaxy functions to update whether a student has attended class, or pull information about location changes, class cancellations, and so forth

checkAttendance(): void

* Retrieve list of daily classes for the current day from the student’s calendar
* Check the notification settings for when the student wants a notification
* Add times that students wants notifications to an array
* Add times that students must be in class
* While loop until the end of the day
  + Run through the list of times you must be in class
  + If the current time matches one of the times in the array of class times
    - Check the student’s current GPS location and match it with the class location
      * If false more than twice for one class, mark as absent
      * If false for first 5 minutes of class, mark Tardy
  + If the current time matches one of the times in the notifications array
    - Send student a notification to go to class

1. Flowchart

For the flowchart I decided to make a diagram of the UML class diagrams for this portion of the code. Much of the GPS tracking system is dependent on member functions of the Student, Calendar, and DailyClass classes interacting. The “main” function per say of this code will have a loop that constantly syncs with galaxy and runs the updateGPS() function and the checkAttendance() function.



1. Data

|  |  |  |
| --- | --- | --- |
| **Name of Data Element** | **Data Type** | **Constraints/Notes** |
| Current GPS | GPS class:  - signed double longitude  - signed double latitude | The student’s/teacher’s current location |
| class | Class object:  - string class name  - Professor class  - class schedule  - string classLocation  - GPS classGPSLocation | Stored in list (containing all classes for a student) |
| professor | Professor class:  - string name  - string title  - string office location  - GPS office location  - Class list  - Date list of office hours |  |
| class schedule | List of date objects:  date object:  - string start time  - string end time  - string day of week |  |
| calendar | Hash Table with every day of the semester stored in Day objects  Day object:  - string key “month/day/year”  - string time zone  - list of dailyClass objects  dailyClass object:  - Class object  - bool list storing whether you’re in class or not  - bool classCanceled |  |