Project Milestone Two

Team: CompOne

Module: Attesa

Anthony Pacitto

Dariusz Kilpinski

Winson Fink

CENG319

September 2, 2018

**Project Milestone Two**

**1. Distribution of Work**

After module one, the team was able to negotiate and distribute tasks evenly among each group member. In order to improve work efficiency, we determined whose strengths fit where. For the java code implantation, we concluded it was best for Anthony to take on the role of the leader and SQLite database control. He is responsible for producing not only the database, but the functionality of its CRUD operations within Java, while also taking on the task of producing concept-only design templates per activity (i.e. via Design Document and basic .xml code). Dariusz has been appointed as the primary lead for a majority of the .xml creation (excluding object id’s since Anthony requires these within Java; thus, Anthony will handle these). Finally, Winson will be managing all the flow between activities making sure that not only are they implemented correctly, but also match with the terms set by the Requirements document. It is important to note, that Dariusz and Winson will work hand-in-hand with the .XML documents once given the concept mockups via Anthony. All team members will be given an equal workload in order to improve efficiency and reduce the chance of error (i.e. too much work may result in critical errors to occur within the project). All alpha testing will occur amongst all three members on multiple platforms in order to assure no platform bugs/errors occur (i.e. the more team members alpha testing at once will increase the odds of finding any existing bugs/error that may exist).

**2. Work Progress**

Currently, the project has made significant progress over the course of the last two weeks. More specifically, we were able to build the most difficult portion of our mobile application: database implementation (i.e. SQLite services). The application can now perform all mandatory CRUD queries between the device and local database. Over the next few days, the team will determine if whether the local database can be expanded further to server-based environment (i.e. ORACLE, SQL, etc.), where all database records can be accessible to multiple users rather than our single device. At the same time, Winson will be programming basic UI using the Styles.xml and activity layouts; while Dariusz and Anthony handle the Java source code (i.e. Anthony expanding database functionality and Darisuz creating the standard Java code for elements that can be put into the project without the need of a database). We have also finalized the concept designs for the overall architecture and graphical theme of the application, in addition to a schematic of the requirements between the various activities and its database. The next phase for our team will be to implement a cleaner GUI, in addition to testing out various algorithm calculations in order to determine which would be the most efficient and accurate method to fetch live times (i.e. this may require a secondary application on the clinic-side which will provide updated patient counts). If needed, a second application would be a very basic application (e.g. counting system per button click) with very minimal time required to construct (i.e. approximately 1 day).

**3. Requirements Analysis**

***3a) User analysis***

The following table indicates the requirements for a given **user** who accesses the application:

|  |  |
| --- | --- |
| **Requirements** | **Stories** |
| 1. Locate the nearest ‘online’ clinics. | The user needs/wants to be able to identify and select clinics whose status is set to ‘online’ only. |
| 2. Ability to select and view information. | The user wants to be able to select additional details regarding a clinic, such as: ratings, reviews, average wait times, real-time wait times, etc. |
| 3. Be able to log into separate accounts. | Each user should be able to create a private account and have access to a secure login system. Each account should hold a personalized rating by each clinic they have visited, as well as a review section to post publicly open comments (i.e. was the patient on time? Thumbs up; thumbs down). |
| 4. Be able to Cancel visits accordingly. | The user must be able to manually cancel a visit while waiting and then be able to see their new rating (i.e. potential feature: each time a user cancels it reduces their rating score). |
| 5. Access to clinic registration forms. | Each user who selects a clinic at a given time, will be provided with all registration forms (i.e. if any exist) prior to visit. |

***3b) System analysis***

The following table indicates the requirements for a given **system**:

|  |  |
| --- | --- |
| **Requirements** | **Stories** |
| 1. Locate the nearest ‘online’ clinics. | The system needs to be able to calculate radius via GPS location in order to collect surrounding clinics within a given range that are open/online only. |
| 2. Manage all clinic and patient details. | The system must store sensitive and encrypted information via both the clinic and patients. More specifically a secure database and CRUD. |
| 3. Secure encryption of sensitive data. | The application data must be automatically encrypted and decrypted on both the client-end and server-end. All patient OHIP values must be secured since they are classified as sensitive data. |
| 4. Be able to log into separate accounts. | A private account module must be accessible by the system. Each account on the system should hold a personalized rating system by each clinic he/she has visited (i.e. was the patient on time? Thumbs up; thumbs down). |
| 5. Live time implementation | The user would like to be able to visually determine the live times of each clinic. Each time range (e.g. 0 – 5min; 10- 15 min; 20-30min, etc.) must be indicated by a color to represent the status of the time (e.g. green -fast, yellow - average, orange - slow, red - very slow) |

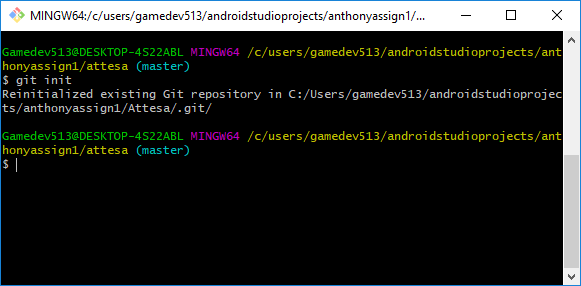
**4. Tool Integration**

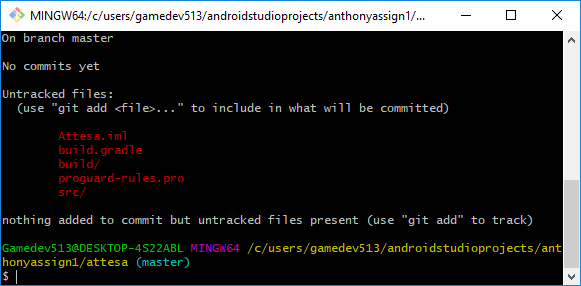
Within the construction and mobile functionality of Attesa, many tools have and will be implemented throughout its development, with some remaining in the final build version 1.0. More specifically, we have primary used the IDE known as Android Studio 3.1 for all java, xml and design development requirements. However, for server-based functionality we integrated the SQLite Database tool to manage all database-functionality and CRUD operations via within in application. In addition, we tend to have access to the GPS location sensor within a Samsung mobile device in order to test radial zone detection. For our icons and GUI images, we have used Photoshop as our main graphic-based tool. Finally, for database manual manipulation and integration, our lead team member has had access to DB Browser for MSQLite (Freeware software provided by sqlitebrowser.org Github.

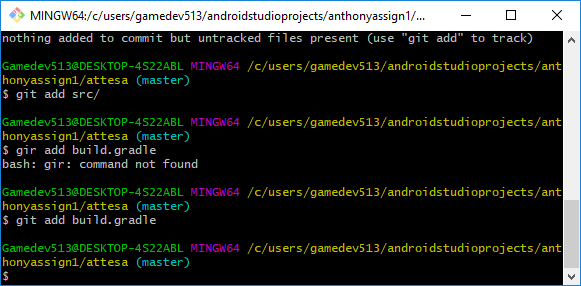
**5. Git and Github Demonstration**

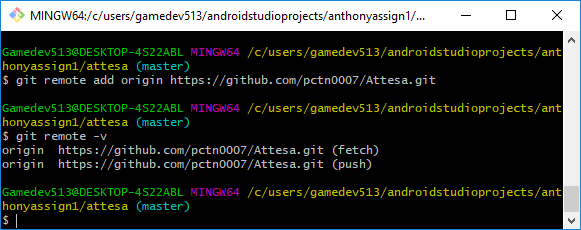
This document will demonstrate an example of how our team members integrate Git with Github in order to modify and fetch files that pertain to our mobile applications. It is important to note that although Git is a powerful tool, all team members felt more comfortable with a public repository on Github since it is known to be very user friendly and easily accessible.

First, we created a new Github repository named after our project application. Next, we used Git terminal to initialize the local directory of the current project as a Git repository, as seem in the first image below. Once we initialized the repository, we verified the status of any files that are not being tracked. You can identify these files by the text color red indicated in the second image below. If we want to track any of these files, we can add them into the repository using the command in the third image. Once we finish adding any new files, we attempted to commit the given files before connecting and pushing to the remote Github repository using the specified URL provided under our Github account repository *Attesa.*









**6. Gantt Chart and Diagrams**

All Gantt, graphical and application-based diagrams are located in additional files within our team folder (PowerPoint and Word documents). Please note Gant chart was developed in Microsoft Projects.