**Android Application**

JustHealth had decided to build its mobile application on the android platform as opposed to any other as their research showed that this was the most widely used system and therefore, would allow us to be able to target as many users as possible. Initially, this seemed to be one of the most daunting parts of the project as although, the group had had some previous experience of Java programming, none of us had any previous experience with using this within an android environment.

Initially, the learning curve was steep although we were able to quickly pick it up through using the android documentation and learning from the mistakes that we had previously made. We set ourselves the task from the outset to implement all of the functionality that we had for the website on the android platform as well, in order to ensure that a user would have a seamless experience independent of whether they were using a mobile, tablet or laptop.

The android application relied heavily on the API that we were developing and therefore, generally we would implement the functionality on the android application in the second week of any given iteration. This gave us chance to be able to implement the API methods that the mobile application would then POST request in order to perform the desired task.

Our biggest achievements with regards to the android application were:

1. Push Notifications using Google’s Cloud Messaging Service (GCM)
2. Integration with Android’s Native Calendar
3. Asynchronous Processing

Firstly, the push notifications functionality was extremely important and a fantastic achievement. This required us to implement methods on the JustHealth server as well as the android application. The building of this functionality was very much an iterative process, firstly we ensured that we was able to retrieve the Registration ID of a given device, a unique ID associated to the specific application and device, this required us to implement methods on the android application. Once we had completed this, we then through the python command line began to try and build an example notification. Once we started to receive the success message from the GCM service provided by Google, we were confident that the notifications were actually being received by the phone although, we were unable to see them - this was due to us not having implemented any methods to display the notification.

Our next step was to implement methods that would show the notification on the phone. Once completed, we built the gcm.py file, which was similar to the api.py although, dedicated to just dealing with the android push notifications. In this file, we implemented methods that would be able to both store and delete the Registration ID of a user’s device, determine the title and the content of a notification and finally push the notification to the relevant device. Lastly, we customised the methods on the android application to ensure that the Registration ID was sent on login, and removed upon logout and also that the notifications were correctly displayed and alerted to the user.

Secondly, the integration of the JustHealth appointment system with the native android calendar was another great achievement of JustHealth’s application. When a user adds an appointment on the android application they have the opportunity to be able to also add this to their native android calendar. If they choose to do this a method automatically adds the appointment to their calendar, the ID of this appointment is then added to the JustHealth database to ensure that it can be referred to at a later date, should the user want to view, update or delete the appointment from the JustHealth application.

Asynchronous processing was key to ensuring that the application was as quick as possible and it was also recommended within the Android documentation. This wasn’t something that we did from the outset and was something that we had to go back to old code and change. However, it gave us the ability to be able to use a loading dialog to show the user that the application was working and hadn’t crashed and also ensure that in places where we had multiple POST requests, on login for example, that we were able to make two at the same time, which sped up the login process.