Xpertise
CS 307 Team 8
Trevor Edris, Josh Foeh, Scott McClellan, Chris Schlatter, Tushar Sharma

What went well?

After we found the Google Cloud API, we were able to successfully create a test application which would connect to a Google Cloud SQL database that we created, retrieve data from it, and then communicate with the GUI and display the retrieved information on the screen. This is part of the basic functionality that we were hoping to ultimately achieve with our application. Setting up the database itself also went quite well as we were able to use the tools provided by the Google Developers Console to create and manage the database. We were also able to test the database using the developers console to ensure that it was working. Since we are using the Google Cloud API to host our backend server, we were also able to directly test the methods we created by using the Google appspot website.

Completed User Stories (for backend):
As a user, I want to make a profile.

For this user story, there were multiple things that we had to accomplish. We had to successfully create a database to store profile information, we had to create a java class to store profile information, we had to create a java class to communicate with the database, we had to create an API class which would provide web-based methods that the Google API would be able to access, and finally we had to create a java class to utilize that API and display the proper information on the GUI (the last of which proved to be a bit difficult).

As a user, I want to login.

We implemented a profile_auth method which will return the profile that matches a given username and password combination. If the profile does not exist it returns null. Since we were unable to get app communications working, this user story was only completed for the backend.

As a user, I want to edit my profile.

We implemented a profile_edit method which will edit the profile of a given pid and return a MyBean object which contains either a success of failure string. Since we were unable to get app communications working, this user story was only completed for the backend.

What did not go well?

We did not manage our time as well as we could have and did most of our work later in the sprint. We became pressed for time and were unable to bring all of the parts together. There were also some issues using the version control software. We did not have very much prior experience with Git and thus had many problems with merge conflicts and getting the project set up initially.
Also, we should have tested the client with the backend API right from the start

Before beginning work on our Sprint 2 tasks, we plan on resolving the following issues:

1. Front end communication with the back end
2. Make the front end more usable and polished

Xpertise

CS 307 Team 8

Trevor Edris, Josh Foeh, Scott McClellan, Chris Schlatter, Tushar Sharma

What can you improve?

We can manage our time better in the future and spread out our work over the entire sprint. This will allow us to deal with unexpected issues more easily as well as allow us to set up more meeting times between specific members if needed.

We can also separate the frontend and backend files into different branches to help alleviate our merge conflict problems. We will also modify our gitignore file to ignore extraneous files so that we have less needless merge conflicts.

If an individual team member requires help, they can ask for it from the other team members more quickly than they did this sprint. We can also have more frequent meetings between individual team members.

Better communication is also necessary from now on. We split up into separate parts with individual responsibilities and ended up having trouble bringing all of the parts together. So we need to communicate basically how the parts work to each other and work together more in making everything work.