Implementation and Testing

MatchMeme

Execution Instructions:
Currently: Download and install Android Studio (https://developer.android.com/studio/). Download our project from our shared Google Drive, or the zipped folder submitted. Plug in an Android device via USB cable. Configure your device to run in Developer mode while connected via USB, depending on the device the way to do this differs (https://developer.android.com/studio/run/device). Follow this guide if you are not sure how to on your device. Once the device is plugged into the computer, run the app. This will cause a pop-up window labeled target device to appear. Your device should appear in this window. Select it and wait for the app to build and run. After a moment the app should appear on your device.
Libraries that need to be included outside of the Android Studio defaults are the Facebook API, specifically the *com.facebook.CallbackManager, com.facebook.FacebookCallback, com.facebook.FacebookException, com.facebook.FacebookSdk, com.facebook.login.LoginResult, com.facebook.login.widget.LoginButton*.

Currently we use Windows 10 devices, however Android Studio works on Microsoft and Apple systems. Android Studio uses a combination of Java and XML languages to make apps. It has an easy to use interface that allows dragging objects onto the screen and then going into the Java and XML behind the objects to further program what they do. Because of this, installing Android Studio, our app and having an Android device is all you should need to execute our project.
Future: Download off of the Google Play Store, (hopefully someday).

User Story Review:
Our user stories are not going to be changed, and we are not adding any new ones. Overall, our user story estimations were accurate, however we underestimated how busy we would be and how much we would be able to get done. This will not change our user stories or their estimates, but we are more aware of the time required to complete this project.

Design Review:
We did not make any changes and are not planning on it. Changes are possible, but we are not planning on changing the base design of the system.

Implementation Review:
The only change we've made to our technology stack is the use of Android Studio, which brings with it XML. This is the only addition to the programming languages portion of the tech stack.

Android Studio would be added to the framework section. Also, we are using SQLite instead of MySQL. Otherwise the rest of our stack will remain the same.

Team Velocity:

Overall, we were very close to our initial estimate for the first sprint provided in part 2. Due to a combination of inexperience, deciding to use a new environment (Android Studio), and a busy last two weeks of school filled with many large final projects, we came very close to having all of our initial goals completed. Of the three goals we had set for the first sprint, we completed the Facebook Login and the majority of the app structure and interface. The design of the app is not quite finished, with layouts and color schemes to still be finalized, and while the meme database is made, it is not yet implemented.

What we each accomplished in our first two week sprint:

Matthew and Evan: We developed the base structure of the app in Android Studio and figured out how to display images and how to switch which image is being displayed. We worked on the layout and design of the app.

Samar: I created a new layout activity for the Facebook login. Originally, we estimated that the level of difficulty for this user story would be a 3. After about 30 minutes of research, I began following the steps provided by Facebook, for developers, to add a “Login with Facebook” button to our application. The instructions provided by Facebook were clear, and therefore the level of difficulty that we initially estimated for this user story was accurate.

Adam: Using the class “MemeDB” created by Lyndon, I attempted to implement the database into our application. However, the code, which was originally tested in Eclipse, was not functional in Android Studio. For this reason, the database was not successfully implemented in the first sprint and the level of difficulty was higher than anticipated.

Lyndon: I created the image database in MySQL using Amazon RDS. To retrieve images from the database, I wrote a function that logs into the database and executes an SQL query which returns image URLs. This function was uploaded to Amazon Lambda and triggered using a REST API generated by Amazon API-Gateway. The API works correctly, returning proper results in Eclipse IDE and web browsers, but causes a network exception in Android Studio. Because of this error, the online database could not be implemented in the first sprint.