Weekly Sprint Notes for OnMyWay

February 20 Weekly Sprint

Team Goal: We will set up the Android Studio project and get the database and maps working to some extent as these are essential for the rest of the project
John: I will set up the database through Firebase, implement an initial user profile and main activity for everyone to use for testing
Neel: I'll get the initial Google Maps API implementation
Bard: I'll add skeleton code for all the classes in the UML so we have a place to start coding from

February 27 Weekly Sprint

Payas + Manpreet: We will implement the rider request activity with search functionality and an activity to generate a fair ride cost for the rider
Bard + Neel: We will find requests within a certain distance (Bard & Neel), add a hamburger menu to interact with profiles and saved requests (Bard), and add polylines and figure out destination handling for the driver side (Neel)
John + Mahin: We will work on implementing input validation, making an activity for the user to edit their profile, the User class, making a DBManager class to handle everything related to Firebase, and making a State class to store objects that are needed across multiple activities in the app (current user, ride request, etc.)
Michael recommends: Make sure you guys are communicating!

March 5 Weekly Sprint

Payas + Manpreet: We will figure out how to implement the request class into a database & add test code and javadocs for rider's side related activities
Bard + Neel: We will add various functionalities to the hamburger menu (Bard), will implement polylines for the rider's side requests (Neel), we will write test code for driver's side activities
John + Mahin: We will work on implementing a sign up activity, a dialog fragment to show the user's profile information, and adding test cases and comments for our code

March 12 Weekly Sprint

Everyone: Test the respective functions you built and work on deliverables, stay in touch on google hangouts so we don't catch coronavirus
Payas + Manpreet: We will continue trying to piece together functionality for the rider's side and implement Firebase support
Bard + Neel: We will work on sending requests back and forth between the driver's and rider's sides, and storing information in the database
John + Mahin: We will work on saving the application state offline, setting up continuous integration with TravisCI, and connecting different parts of the app together

March 26 Weekly Sprint

John : Work on saving the application state offline, setting up continuous integration with TravisCI, and connecting different parts of the app together. For next week, will work on fixing bugs, improving UI, and connecting activities/fragments together for RiderMapActivity. Add fragments for showing the current request for both driver and rider.
Manpreet: Addressed issue(s) regarding the coding of the RiderMapActivity, and added error-handling measures. Will now be working on confirming the completion of a request, and adding the ability to pay with this request. Adding error-handling, and testing in this area as well. Updating UML and Sprint Notes.
Mahin: Working on QR code integration. Adding tests and cleaning up redundant code.
Bardia: Improving view requests to handle with database requests, cleaning up bugs, testing. We usually discuss what we will work on after weekly meeting, will update then.
Payas: Integrated price estimate into RiderMapActivity. The rider will now be shown a price estimate once they confirm a ride and they can offer more money. Will be working on QR bucks for the next week.
Neel: Testing rider and driver map activities. Will work on incomplete user stories for the upcoming week

April 2 Weekly Sprint

John: Writing test cases and comments for the code that was completed last week
Manpreet: Worked on the UML, edited code as needed. Required assistance to complete tasks from previous week.
Mahin: QR integration complete, rating system implemented. Cleaned up redundant code and worked on removing bugs from the code.
Bardia: Driver accepting request to database, listener on rider side that handles the accepted request (fragment), and completed request. Made object that allows us to store requests in google map markers for retrieving from database. Taking out hard coding for view profile fragments, now will view matched person.
Payas: Set status to cancelled in the database when rider cancels ride. Update the map on driver’s side when a new request is added to the database. Comment code.
Neel: Added DriverHistory, refactored RiderMap code, tests to database, updates to DriverMap