

## Android Homework-1 Report

### NYU – Poly CS9033 Mobile Application Programming

Professor: Jeffrey Bickford <jbickford@nyu.edu>

**Controller, Models and Views** were the main focus of this homework which were implemented in the following way:

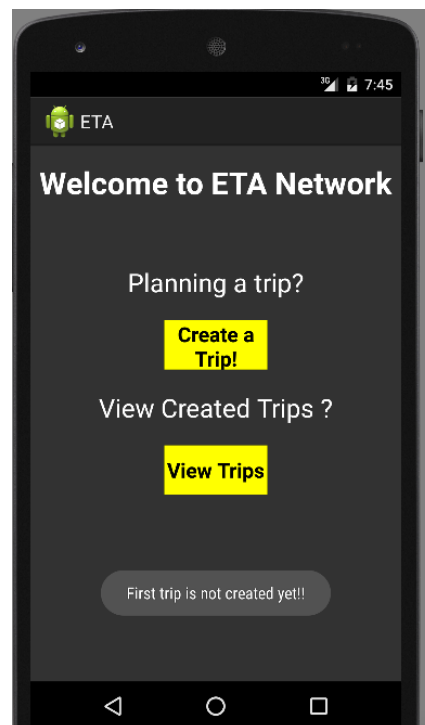
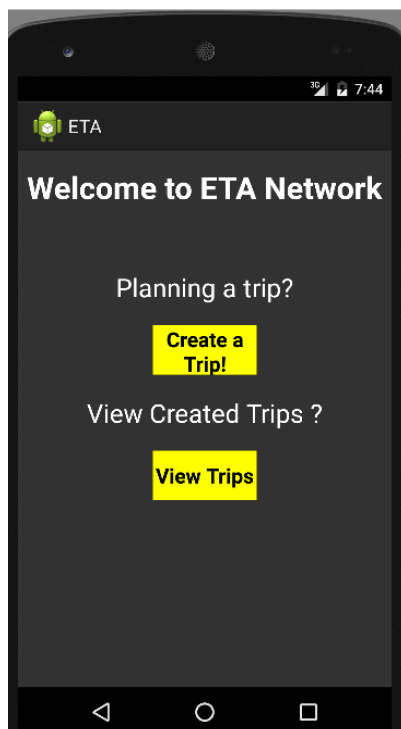
**Views:** The three views implemented are as follows:

1. **activity\_main:** Main activity of the app. Whenever the app is used via the launcher this activity gets displayed. This is the default page of the app where the finish () step in the cycles always finishes.

In screenshot below we can see there are three defaults texts and two relating buttons which would lead to the other two activities. The texts help us in selecting the options if using as a novice.

**Create Trips Button:** Navigates through the us to the activity\_create\_trip and helps creating a new trip by filling up the various details.

**View Trip Button:** Navigates us to the activity\_view\_trip and the user is able to see the latest trip details that has been saved by the user.



The mechanism of the activity is that user can create as many trips as he wants to when he clicks on the create trips , and the user is not able to see the latest trip if there are not trips created by then. User has to create a trip in order to view one.

**2.activity\_create\_trip:** User lands up on this activity whenever he clicks on the Create Trip button. We can fill in the details on the trip with various fields like :

Name of the trip:

Meeting Location:

Trip Destination:

Trip Time:

Trip Date:

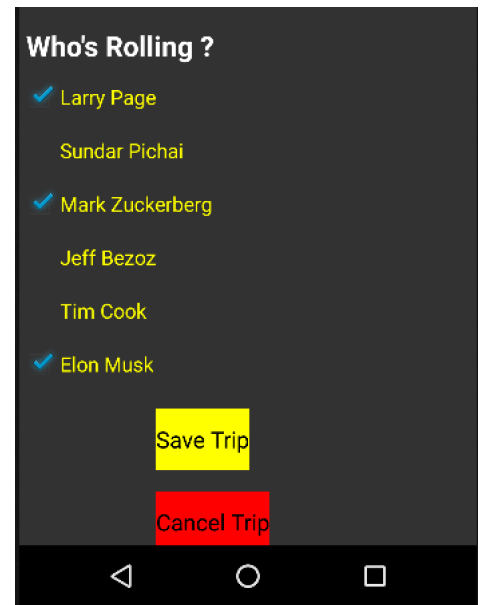
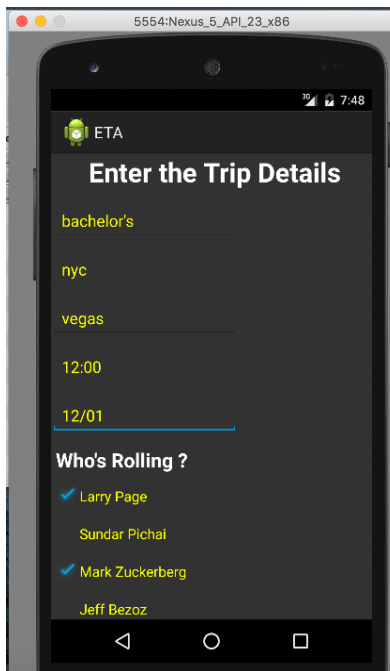
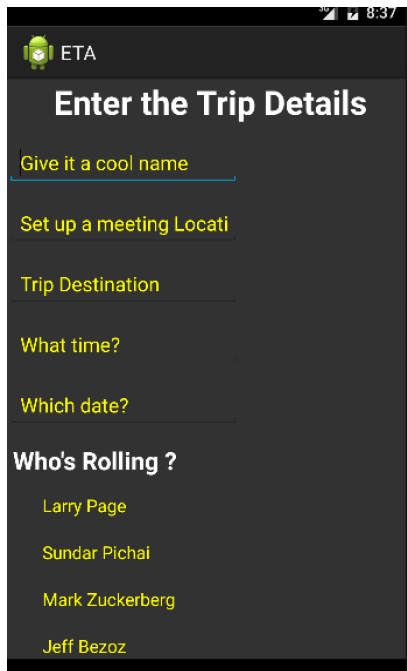
Attendees:

User is allowed to check mark certain number of friends with the help of check boxes he wants to invite for the trip from a list of the friends in this network and only those friends will be considered for the trip.

All the other fields have been given a hint name to help the user fill in the details for each field. Also the scroll view is enabled to view the every fields and the list of friends and the buttons at the end.

**Save Trip Button:** It saves the details of the trip and redirects back to the activity\_main view and when the user clicks on the View Trip Button one is able to see the details entered in that.

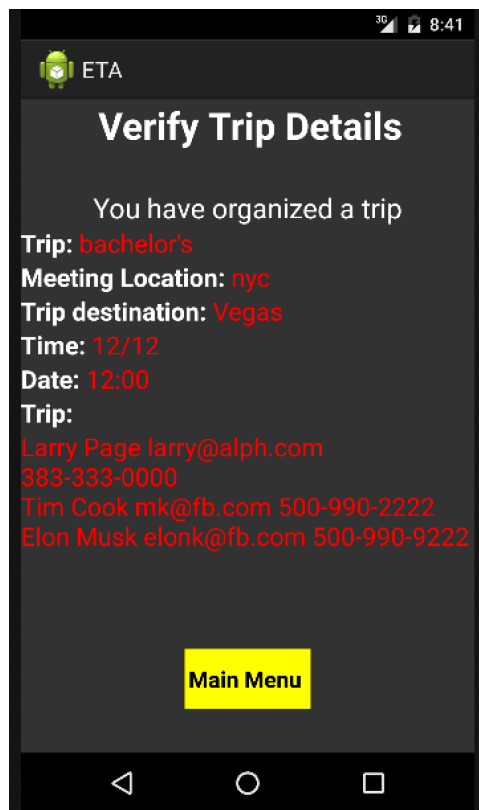
**Cancel Trip Button:** It cancel the trips and does not save the details of the trip and redirects back to the activity\_main view while the same can also happen with the back press from the phone.



3. **activity\_view\_trip**: The user is able to view the details of the trip which he had created last. The different fields are displayed with the labels indicating what values signify what meaning.

Also the list of the friends that are attending the event are also listed along with the email addresses and the contact numbers for the same thus the user is able to contact them easily relating to his trips.

**Main Menu Button** : This button redirects to the activity\_main when the user is finished viewing the details for the trip and thus it completes the entire flow of the app.



## Models:

There are two models created as follows

**Person:** The details of the friends are extracted via this class and presently name, email and phone numbers of the classes are included in the information provided through them. This class implements parceable interface so that the objects can be serialized.

**Trip:** This model is created for storing the details of the trip to be created. The classvariables used are as follows: name, location, date, time, meetingLocation, friends1. This class implements the interface Parceable so that the object can be serialized.

**Controllers:** Following are the three controllers used:

**MainActivity.java:** In this controller we pass the control of the activity\_main to the

activity\_create\_trip or activity\_view\_trip depending on the button pressed. When create trip button is pressed we use the startActivity method as we are expecting some value from the activity\_create\_trip.

While navigating to the activity\_view\_trip if we find that Trip is null then we pop a Toast saying that No trip is created. If Trip is not null we navigate to activity\_view\_trip.

**CreateTripActivity.java:** In this controller we create a Trip object from the values entered in the EditTexts on the view. If any value is missing then we pop a Toast that 'No field can be left blank'. If all fields are entered then we create the Trip object and pass it to the activity\_main.

**ViewTripActivity.java:** In this controller we extract the Trip details from the trip object and display it on the desired fields.