DYNAMIC GENERATION OF PUBLIC TRANSPORT INFORMATION

INITIAL PLAN

John William Olegario
C1429814 – CM3203 One Semester Individual Project - 40
School of Computer Science & Informatics

Supervised by: Dr. Martin Caminada

Moderated by: Dr. Federico Cerutti

25 January 2018

Table of Contents

Project Description

Tourist travelling on countries with less internet connection is a challenge. Faced with questions such as *where do I catch the bus? Reliable transport information? Nearest terminal? Where do I stop to reach my destination? Nearest restaurants? Restrooms?* In the western world countries such as the United States, United Kingdom, Netherlands or Luxembourg where internet access is accessible wherever or whenever to aid on answering some of the questions mentioned above with use of travel planner applications such as *Traveline Cymru* in Wales, *9292OV* in the Netherlands, or the *Mobiliteits Zentral* in Luxembourg. In contrast to countries such as Malaysia or the Philippines where the internet connection is less or there is simply no application to help tourists travelling the country to aid them on locating or obtaining the crucial information they need when visiting a different country. Asking the locals within the country these questions is not generally a problem, however there are many barriers that exists when this happens such as language e.g. They cannot speak English or Dutch, Disturbance e.g. Nuisance of stopping locals to ask questions, Internet e.g. Wait until a Wi-Fi area is discovered, delaying tourists from acquiring relevant information quickly. TARA is a Filipino word for *let us go*, which is the chosen application name.

TARA is a travel companion application in Android (Google) that aims to aid with this problem one that dynamically collects information from a travelling user base such as tourists. In instance, when travelling from *Puerto Princessa Palawan Philippines* the application would detect that the user has left the departure terminal by looking at location, speed and time. A typical low-speed vehicle can run up to 25+ mph (40 km/h), at this rate the application will assume that the user is inside a fast-moving vehicle, at 10 -20 mph is considered a slow-moving vehicle and 0 – 9 mph is considered as a human manageable speed when walking or running. The application will consider when it came into a full stop with a short delay of 3 minutes before the application assumes that the user has reach its destination otherwise if considerable amount of speed is gained then the user has not reached its destination. This speed and time data will be eventually automatically stored or updated in a NOSQL MongoDB database server. Supplement information are optionally added by users which was possible due to the creation of user accounts that will use social media elements that enables users to post statuses e.g. *busses do not run on weekends* or images e.g. *bus timetable*. Application will also ask user input if there are no internet connection before and after travel.

The social media elements of the application are purposely included as the back-end data will continuously analyse the data which will be used to provide useful information to other incoming, returning or current tourists information about the certain location or country that they are about to visit or have visited. Users can post statuses and reply to other user's statuses. Not to be mistaken with social media sites such as Facebook or Twitter, this application is a mechanism that enables users find the information with the aid of social media aspects that ensures users find more information from other users based on their statuses.

In addition, quick tap icons will reveal immediate view of transport information such as bus or terminal travel information which enables users in finding how to get to certain locations if it is available, search query boxes will be available in the user-interface with google maps access.

Project Aims & Objectives

The overall aim is to provide a dynamic android application companion to a travelling user base one that provides immediate and relevant information.

No internet connectivity is a massive consideration during the planning and implementation of TARA, which justifies the use of NoSQL database MongoDB due to its BASE (**B**asically **A**vailability, **S**oft state, **eventually**) consistent approach when storing and updating data. The information on display to the user is consistent if they have internet connection because the application will automatically deduce information using the GPS and Google maps data in identifying the user's location. When there is no internet these features may not be available, and information will be inconsistent and when this occurs, the application will rely on users input or at least before travel the user was in a Wi-Fi secured place. This inconsistency is considered as acceptable as it is only temporary, the database will eventually be updated with the latest data as soon as internet connectivity is resumed which in turn bring updated information into the client android interface, this includes updates from other users which is viewable on a shared statuses view.

Non – Functional Requirements

Usable and easy to navigate the dynamic approach of the application will ensure constant and immediate information display to users with use of icons to quickly tap and query the most relevant information.

Portability Downloadable and installable in many android mobile phone devices.

Availability of the data – Due to less internet connectivity assumption the application considers eventual consistency which results in a temporary inconsistent data until new data is successfully stored in the database ensuring data are always available to users.

Dynamic – The application changes it state constantly from online to offline mode and one that ensures the updates information constantly as soon as it is saved or updated data in the back end which provided new information in the front end.

Must Have Requirements

- New users will be able to create a new account requiring their first name, last name and email address only.
- Existing users will able to login using a username and password.
- Google maps icon when tapped assuming internet connection is available will automatically determine user's location.
- When internet connection is not available, users must have the ability to input location manually.
- User's ability to post a status and take a photo as part of posting a status.
- Bus icon when tapped by the users will reveal ask users to find a destination and the app will suggests the possible best direction or guidance based on the location of the user.
- Constant data collection from the user, which are either a manual input or automatically generated based on internet connectivity, based on location, time and speed.
- User account database are associated to the post, images and generated data (*location, speed, time)*.
- Query of information.

Should Have Requirements

- Dynamic change of icon colour green to grey informing user when it is online or offline mode.
- Dynamic appear and disappearance of bus icon and automatic delay with a refresh icon with interval of 3 minutes or more in determining whether a user is inside a vehicle or not with additional input from the user for confirmation.
- Recent status view of other users posts in a timeline manner based on the location of the user.
- Search text box giving the users the ability to query further information if data is available e.g. *nearest restaurant.*

Desirable Requirements

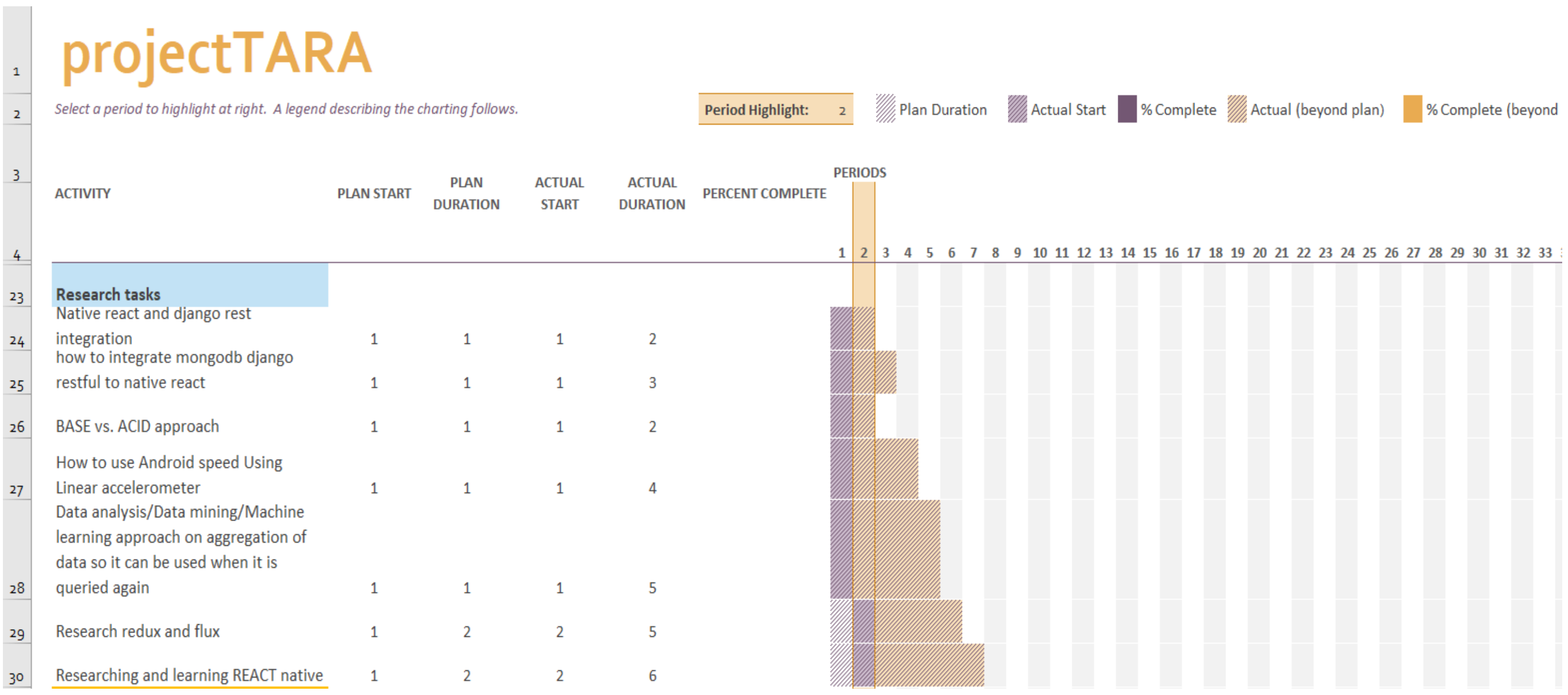
- Data mining / Machine learning ability that analyses user statuses and rates the good and bad options when travelling to a destination, advises users the good options. E.g., *it is better to take the ferry than bus.*

Ethical aspects

During the testing phase, the application requires human participants and acquiring personal or sensitive data such as email address, location which will be stored into a database for analysis. Due to this, ethical guidelines will be followed set by the Cardiff University including submission of an ethical form to ensure the project conforms within the data protection laws.

Gantt chart

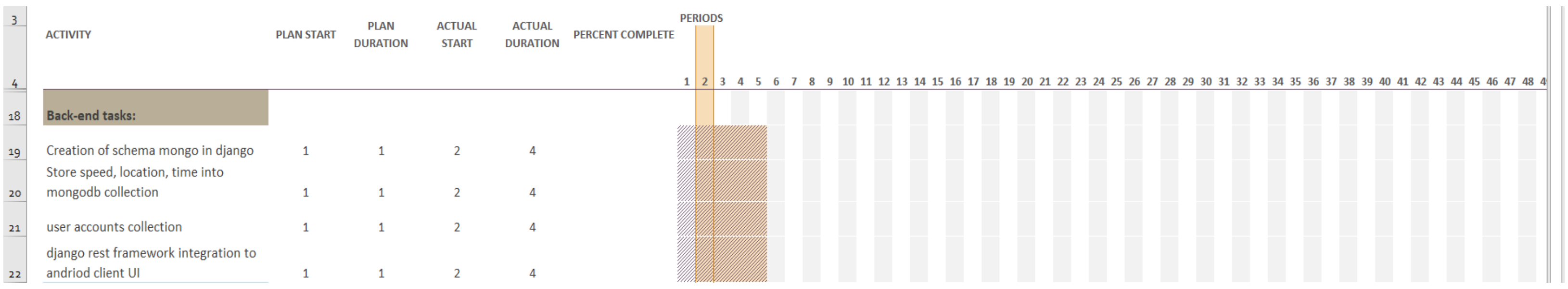
Plan is subject to change at any time



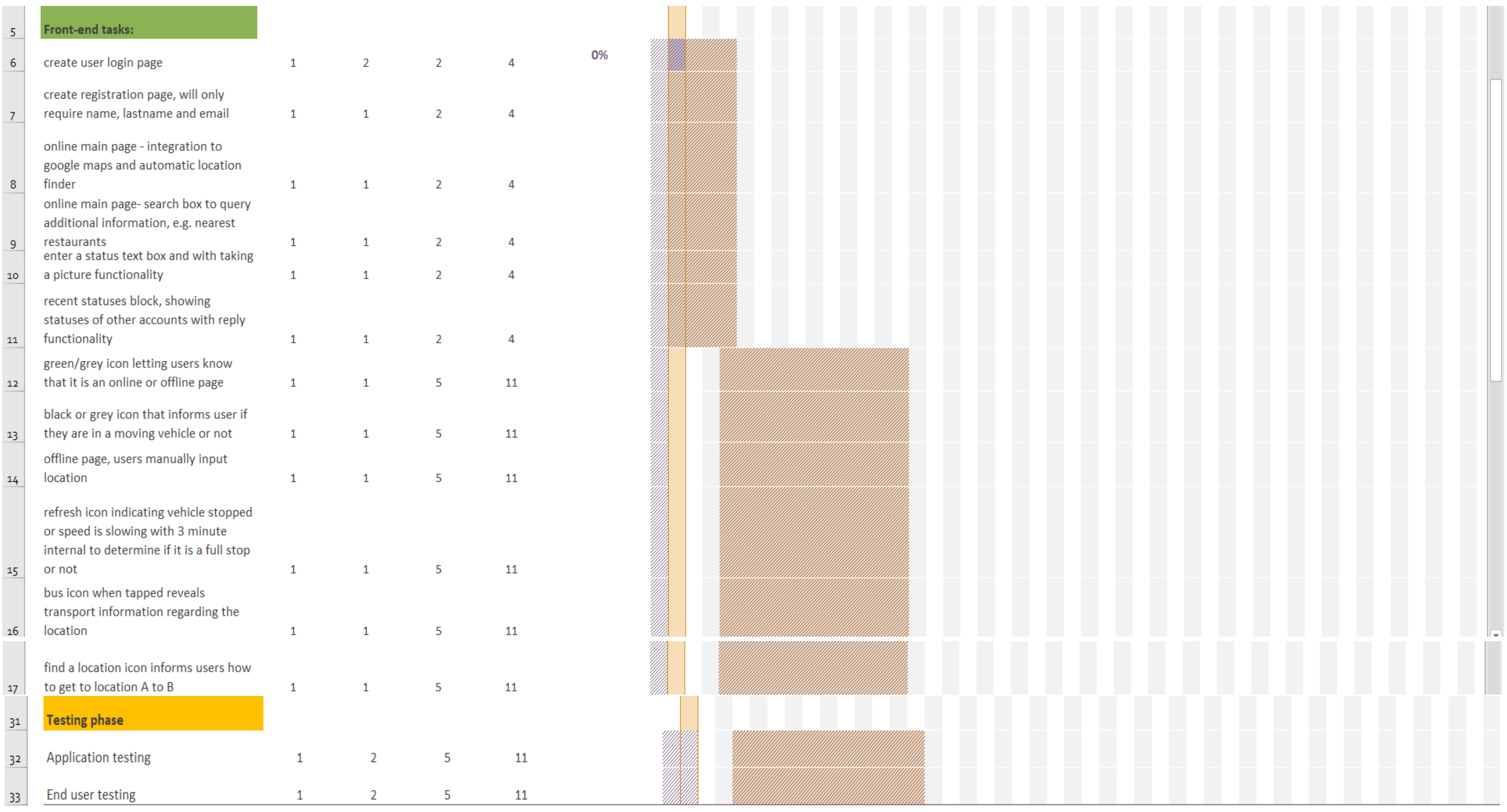
projectTARA

Select a period to highlight at right. A legend describing the charting follows.

	ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
23	**Research tasks**					
24	Native react and django rest integration	1	1	1	2	
25	how to integrate mongodb django restful to native react	1	1	1	3	
26	BASE vs. ACID approach	1	1	1	2	
27	How to use Android speed Using Linear accelerometer	1	1	1	4	
28	Data analysis/Data mining/Machine learning approach on aggregation of data so it can be used when it is queried again	1	1	1	5	
29	Research redux and flux	1	2	2	5	
30	Researching and learning REACT native	1	2	2	6	

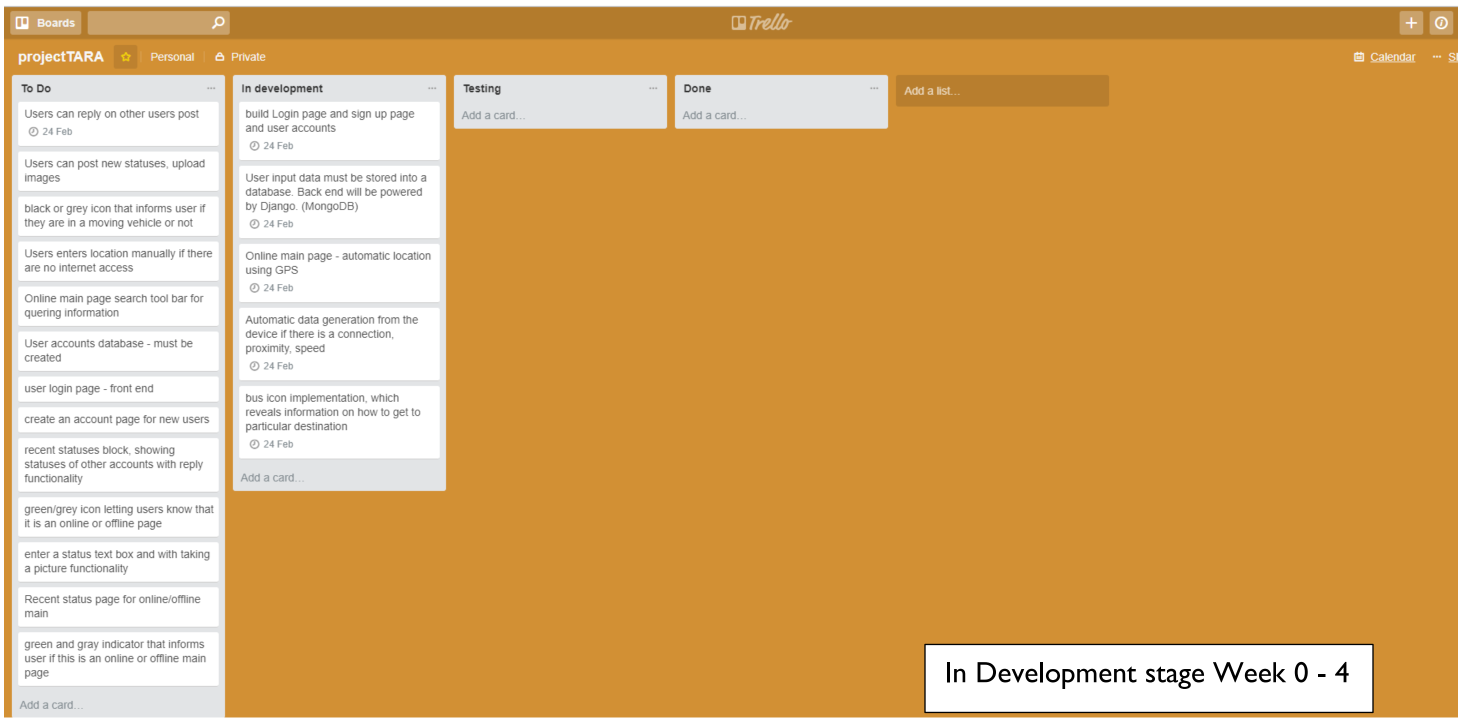


	ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
18	**Back-end tasks:**					
19	Creation of schema mongo in django	1	1	2	4	
20	Store speed, location, time into mongodb collection	1	1	2	4	
21	user accounts collection	1	1	2	4	
22	django rest framework integration to andriod client UI	1	1	2	4	



5	**Front-end tasks:**					
6	create user login page	1	2	2	4	0%
7	create registration page, will only require name, lastname and email	1	1	2	4	
8	online main page - integration to google maps and automatic location finder	1	1	2	4	
9	online main page- search box to query additional information, e.g. nearest restaurants	1	1	2	4	
10	enter a status text box and with taking a picture functionality	1	1	2	4	
11	recent statuses block, showing statuses of other accounts with reply functionality	1	1	2	4	
12	green/grey icon letting users know that it is an online or offline page	1	1	5	11	
13	black or grey icon that informs user if they are in a moving vehicle or not	1	1	5	11	
14	offline page, users manually input location	1	1	5	11	
15	refresh icon indicating vehicle stopped or speed is slowing with 3 minute internal to determine if it is a full stop or not	1	1	5	11	
16	bus icon when tapped reveals transport information regarding the location	1	1	5	11	
17	find a location icon informs users how to get to location A to B	1	1	5	11	
31	**Testing phase**					
32	Application testing	1	2	5	11	
33	End user testing	1	2	5	11	

Work plan



To Do

- Users can reply on other users post — 24 Feb
- Users can post new statuses, upload images
- black or grey icon that informs user if they are in a moving vehicle or not
- Users enters location manually if there are no internet access
- Online main page search tool bar for quering information
- User accounts database - must be created
- user login page - front end
- create an account page for new users
- recent statuses block, showing statuses of other accounts with reply functionality
- green/grey icon letting users know that it is an online or offline page
- enter a status text box and with taking a picture functionality
- Recent status page for online/offline main
- green and gray indicator that informs user if this is an online or offline main page

Add a card...

In development

- build Login page and sign up page and user accounts — 24 Feb
- User input data must be stored into a database. Back end will be powered by Django. (MongoDB) — 24 Feb
- Online main page - automatic location using GPS — 24 Feb
- Automatic data generation from the device if there is a connection, proximity, speed — 24 Feb
- bus icon implementation, which reveals information on how to get to particular destination — 24 Feb

Add a card...

Testing

Add a card...

Done

Add a card...

Add a list...

In Development stage Week 0 - 4

Week 0 (22 January – 27 January)

- Learn React native further
- Begin to produce application flow chart and finalise the relevant workflow of the application
- Begin to produce mock ups design using Balsamiq to finalise user interface design of the application
- Research datamining/machine learning classifiers for good and bad, neutral detection of terms
- Complete ethical course in Learning Central.

Week 1 (29 January – 3 February)

- Consult with the supervisor to review final mock up design and workflow and provide finalised clarifications on how we want the application to work.
- Submit ethical form.
- Begin creating user log in and sign up page.
- Begin creating Django application rest framework and integrate MongoDB.
- Consult with IT if back-end Django rest API support is available within the school web servers.
- Consult with Catherine Teehan for any spare raspberry pi.
- Complete initial plan.

Week 2 (5 February – 10 February)

- Research react native and Django rest framework integration.
- Research state and stateless cases in react native.
- Begin creating application main page.
- Begin implementing automatic location detection functionality.

Week 3 (12 February – 17 February)

- Research how to use Android linear speed accelerometer.
- Begin creation of the MongoDB collection that accommodates user accounts associated with attributes such as email, speed, location, and time, post, reviews, images.
- Begin bus icon implementation.

Week 4 (19 February – 24 February)

- Begin implementation of speed detection.
- Begin implementing search functionality.

Milestones: Partial completion of **must have** requirements.

Week 5 (26 February – 3 March)

- Implementation of user's ability to post status and reply to another user's status.
- Implementation of recent status view.

Week 6 (5 March – 10 March)

- Application testing of the application for the partially completed requirements.
- Continue development of features that are not finished.
- Bug fixes.

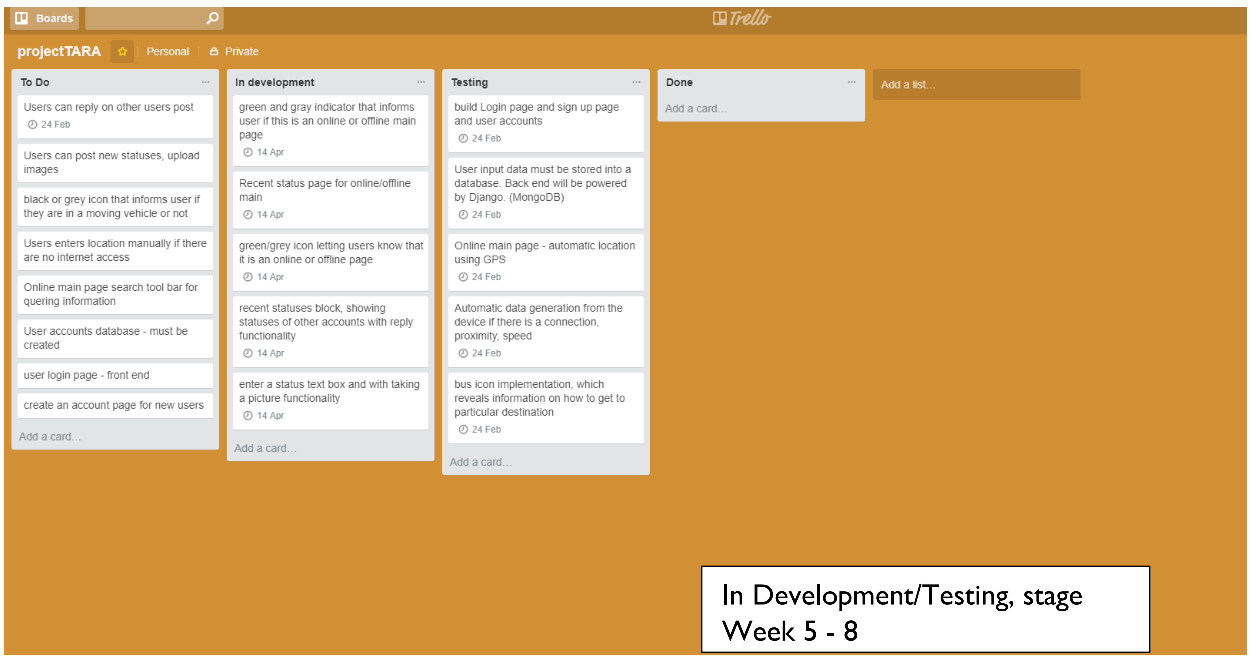
Week 7 (12 March – 17 March)

- Bug fixes.
- Begin development of should have requirements.
- Begin development of desirable requirements *(optional)*.

Week 8 (19 March – 24 March)

- Bug fixes.
- Continue development of should have requirements.
- End user testing in Cardiff.

Milestones: Completion of **must have** requirements, end user testing in Cardiff is tested towards must have requirements.



To Do

Users can reply on other users post
24 Feb

Users can post new statuses, upload images

black or grey icon that informs user if they are in a moving vehicle or not

Users enters location manually if there are no internet access

Online main page search tool bar for quering information

User accounts database - must be created

user login page - front end

create an account page for new users

Add a card...

In development

green and gray indicator that informs user if this is an online or offline main page
14 Apr

Recent status page for online/offline main
14 Apr

green/grey icon letting users know that it is an online or offline page
14 Apr

recent statuses block, showing statuses of other accounts with reply functionality
14 Apr

enter a status text box and with taking a picture functionality
14 Apr

Add a card...

Testing

build Login page and sign up page and user accounts
24 Feb

User input data must be stored into a database. Back end will be powered by Django. (MongoDB)
24 Feb

Online main page - automatic location using GPS
24 Feb

Automatic data generation from the device if there is a connection, proximity, speed
24 Feb

bus icon implementation, which reveals information on how to get to particular destination
24 Feb

Add a card...

Done

Add a card...

In Development/Testing, stage Week 5 - 8

Week 9 (26 March – 31 March)

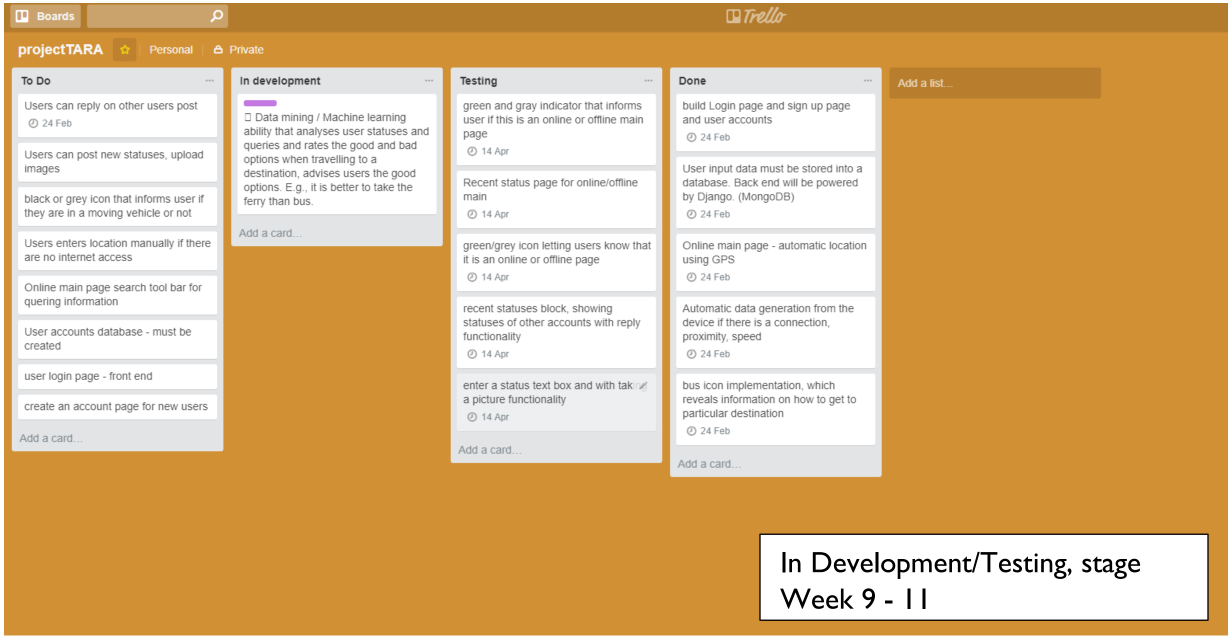
- Continue development of should have requirements.
- Bug fixes.
- End User testing in Derby.

Week 10 (2 April – 7 April)

- Continue development of desirable requirements *(optional)*.
- Continue development of should have requirements.
- Bug Fixes.
- Begin final report documentation.

Week 11 (9 April – 14 April)

- Bug fixes.
- Continue final report documentation.
- Continue development of desirable requirements *(optional)*.
- Application testing.
- Secondary End user test in Cardiff.



To Do

- Users can reply on other users post — 24 Feb
- Users can post new statuses, upload images
- black or grey icon that informs user if they are in a moving vehicle or not
- Users enters location manually if there are no internet access
- Online main page search tool bar for quering information
- User accounts database - must be created
- user login page - front end
- create an account page for new users

In development

- □ Data mining / Machine learning ability that analyses user statuses and queries and rates the good and bad options when travelling to a destination, advises users the good options. E.g., it is better to take the ferry than bus.

Testing

- green and gray indicator that informs user if this is an online or offline main page — 14 Apr
- Recent status page for online/offline main — 14 Apr
- green/grey icon letting users know that it is an online or offline page — 14 Apr
- recent statuses block, showing statuses of other accounts with reply functionality — 14 Apr
- enter a status text box and with taking a picture functionality — 14 Apr

Done

- build Login page and sign up page and user accounts — 24 Feb
- User input data must be stored into a database. Back end will be powered by Django. (MongoDB) — 24 Feb
- Online main page - automatic location using GPS — 24 Feb
- Automatic data generation from the device if there is a connection, proximity, speed — 24 Feb
- bus icon implementation, which reveals information on how to get to particular destination — 24 Feb

In Development/Testing, stage
Week 9 - 11

Milestones: Completion of **must have & should have** requirements and partial completion of desirable *(optional)*.

Week 12 (16 April – 21 April)

- Continue final report draft.
- Bug fixes.
- Continue development of desirable requirements *(optional)*.

Week 13 (23 April – 28 April)

- Continue final report draft
- Bug fixes
- Continue development of desirable requirements *(optional)*.

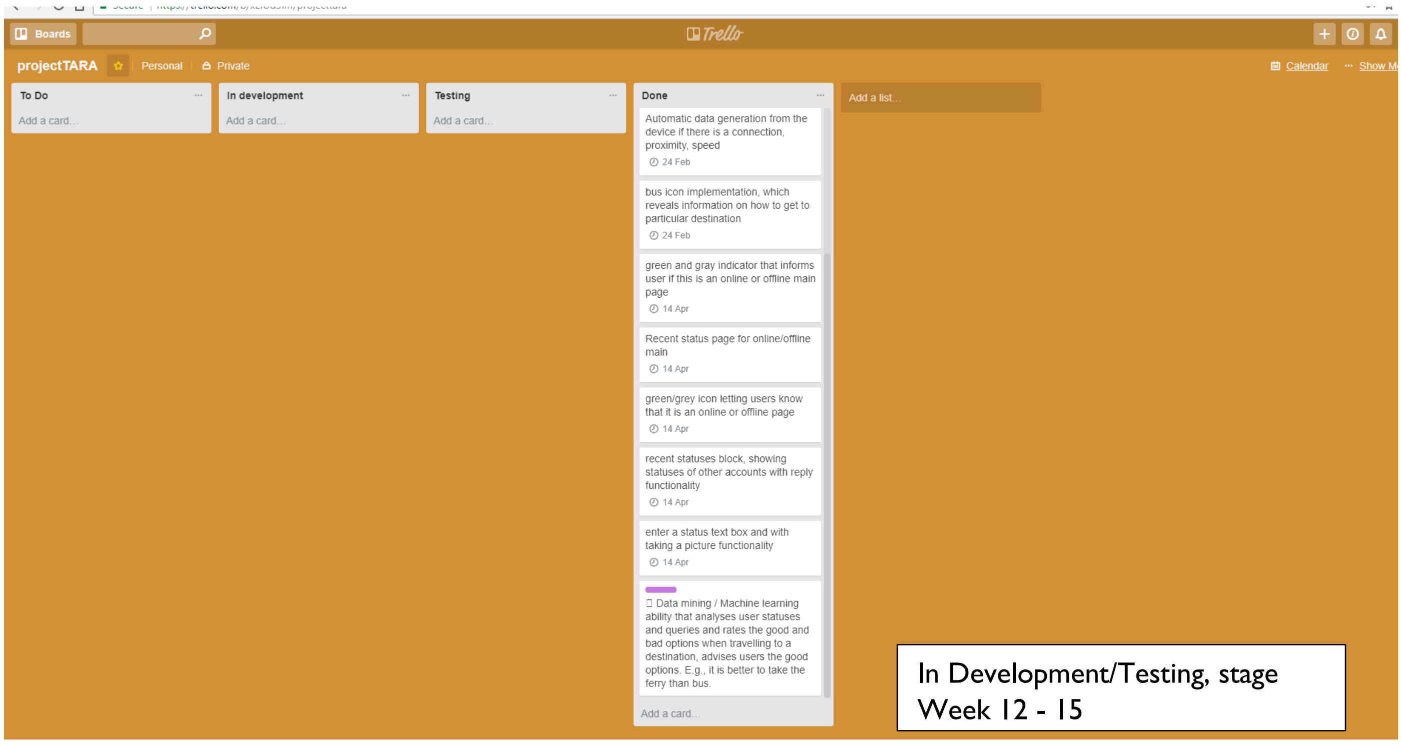
Week 14 (30 April – 5 April)

- Final check-up and proofread of the final report

Week 15 (7 May – 11 May)

- Submission of final report

Milestones: Partial or Completion of desirable requirements (*optional)* and submission of the report.



To Do

In development

Testing

Done

- Automatic data generation from the device if there is a connection, proximity, speed — 24 Feb
- bus icon implementation, which reveals information on how to get to particular destination — 24 Feb
- green and gray indicator that informs user if this is an online or offline main page — 14 Apr
- Recent status page for online/offline main — 14 Apr
- green/grey icon letting users know that it is an online or offline page — 14 Apr
- recent statuses block, showing statuses of other accounts with reply functionality — 14 Apr
- enter a status text box and with taking a picture functionality — 14 Apr
- □ Data mining / Machine learning ability that analyses user statuses and queries and rates the good and bad options when travelling to a destination, advises users the good options. E.g., it is better to take the ferry than bus.

In Development/Testing, stage
Week 12 - 15

Bibliography

PYPI (2018) *Django mobile-app-distribution 0.5.* Available from: https://pypi.python.org/pypi/django-mobile-app-distribution accessed 24 January 2018

Scotch (2018) *Creating a MEAN Stack Google Map App.* Available from: https://scotch.io/tutorials/making-mean-apps-with-google-maps-part-i accessed 24 January 2018

React Native (n.d.) *Building native mobile apps using JavaScript and React.* Available from: https://facebook.github.io/react-native/docs/signed-apk-android.html accessed 25 January 2018

Medium (2017) *Creating native apps with Django rest-API.* Available from: https://medium.com/@hassanabid/creating-react-native-apps-with-django-rest-api-59e8417865e9 accessed 25 January 2018

Edgecoders (2017) *The difference between Flux and Redux.* Available from: https://edgecoders.com/the-difference-between-flux-and-redux-71d31b118c1?gi=800e64be88ce accessed 28 January 2018

Moyer H | Mew K (2016) *Android Application Development Cookbook – Second Edition*

Microsoft Azure (n.d.) *Introduction to Azure Cosmos DB: API for MongoDB* Available from: https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb-introduction accessed 29 January 2018

Chris Umbel (2016) *Using MongoDB as a Backend for Django with django-mongodb-engine* Available from: http://www.chrisumbel.com/article/django_python_mongodb_engine_mongo accessed 29 January 2018

DataQuest (2015) *Naïve Bayes: Predicting movie review sentiment* Available from: https://www.dataquest.io/blog/naive-bayes-tutorial/ accessed 1 February 2018

Smashing Magazine (2014) *An introduction to Node.js and MongoDB* Available from: https://www.smashingmagazine.com/2014/05/detailed-introduction-nodejs-mongodb/ accessed 1 February 2018

OpenShift (2012) *Rapid Python and Django App Deployment to the Cloud with PaaSBlogs:Quickstarts* Available from: https://blog.openshift.com/rapid-python-and-django-app-deployment-to-the-cloud-with-a-paas/ accessed 2 February 2018

StreamHacker (2010) *Text classification for sentiment analysis – naïve Bayes classifier* Available from: https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/ accessed 2 February 2018