

# COM1001 Team Project: ISE 2019

## Team 29's Report - Alta Taxis

<b>Members and teamwork</b>	<b>1</b>
<b>Twitter info</b>	<b>1</b>
Sheffield admin login	2
Manchester admin login	2
<b>System setup before running (very important)</b>	<b>2</b>
Clone the project git repository	2
All other requirements	2
How to run the app	2
<b>Walkthrough the app</b>	<b>3</b>
Welcome page	3
Login	3
My account	4
Admin dashboard	4
Analytics	4
Twitter marketing	5
Admin settings	5
Logout	5
<b>Story cards</b>	<b>5</b>
Story cards removed	5
First iteration story cards	6
Second iteration story cards	8
Story cards changed from first iteration	8
Unchanged story cards tackled in this iteration	10
Story cards added from first iteration	11
<b>Story card burndown chart</b>	<b>12</b>
<b>Testing</b>	<b>14</b>
Manual tests	14
The automated tests	16

# Members and teamwork

Below is the list of the members of our group:

- Chan Ken Lok
- Kassandra Yuen
- Tom Tonner (AquinasAhoy for some of the commits)
- Manas Sarpatwar
- Simonas Petkevicius
- Ibrahim Sinan Bicer
- Su Zhong Ying

Most of the work was split up by deciding what role would be best suited to each individual.

- Chan Ken Lok was allocated tasks mainly to do with the database and back-end code such as validation.
- Kassandra Yuen designed most of the 'index' and 'my account' pages and implemented the CSS and HTML to make it all work.
- Tom Tonner mainly helped with back-end coding with the admin settings, but also helped in other areas such as minitesting, debugging and writing the reports and story cards.
- Manas Sarpatwar was the main developer who designed and coded the dashboard, as well as the one who created all the cucumber tests.
- Simonas Petkevicius' role was to do the marketing and Twitter analytics pages you see on the admin dashboard.
- Ibrahim Sinan Bicer's main role was manual testing and reporting errors in the system.
- Su Zhong Ying helped with the manual testing and other small tasks to make sure the end product was worthy of distribution and stratified our clients.

In the first iteration most of our work was done on weekends and during meetings where every member was present and worked on the code together, helping each other and communicating in person.

During the second iteration however, we spent most of the time doing work at home and communicating over slack and facebook. This change in work was largely due to the Easter break.

## Twitter info

For you to have admin access in our web-application you'll need to login under our Twitter account. As well as this, there are two seperate admins: the Sheffield admin, and the Manchester admin.

For this access, sign in through Twitter on our application using the respective information below:

## Sheffield admin login

Username: ise19team29 (email: tbtonner1@sheffield.ac.uk, phone: 07500090833)

Password: SoftEng2019

## Manchester admin login

Username: altataxis (email: mmsarpatwar1@sheffield.ac.uk)

Password: SoftEng2019

*Note: Email may be needed to sign in if username fails. And as an admin you can change other user permissions in the admin settings to effectively make other people have various admin access: 0 = normal user, 1 = Sheffield admin, 2 = Manchester admin - in the 'user type' field.*

## System setup before running (very important)

Make sure to follow each step exactly as stated in the following instructions before trying to run our application.

## Clone the project git repository

Open the terminal and in the terminal type:

```
git clone git@git.shefcompsci.org.uk:com1001-2018-19/team29/project.git
```

And then enter your username and password for gitlab.

## All other requirements

For all other requirements for our system, first go into project directory (using `cd`) and run the following commands in the terminal:

```
rake installrvm
```

Then:

```
source /home/codio/.rvm/scripts/rvm upgrade 2.6.0
```

Next:

```
rake install
```

*Note: this will take a while and requires two confirmations to complete the process*

And finally:

```
rake addcallback
```

## How to run the app

Once all requirements are installed you can start running the app by making sure you're in the project directory (using `cd` if needed) and in the terminal running the command:

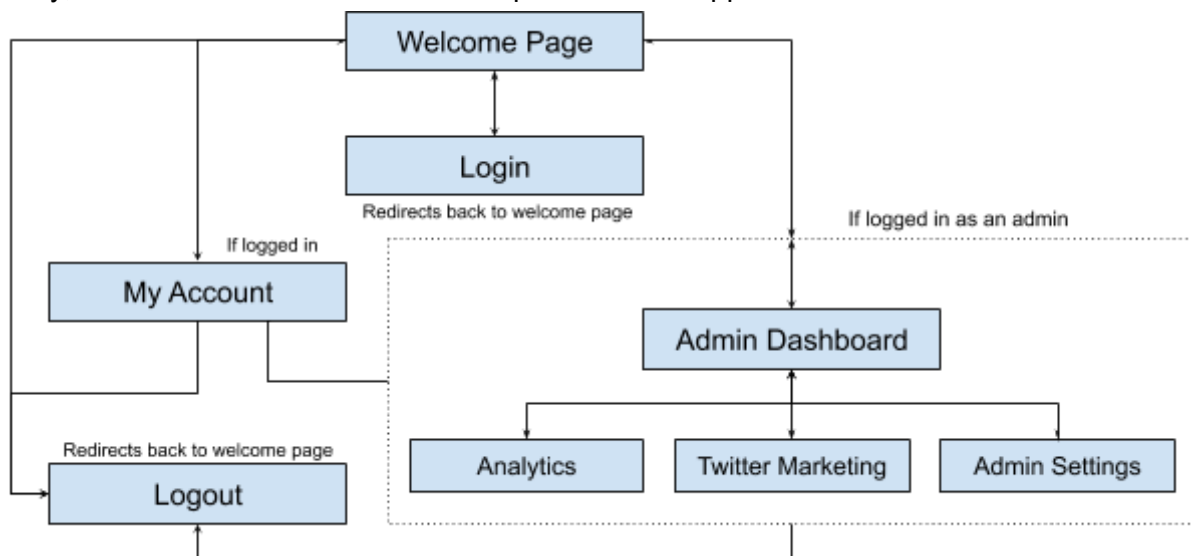
```
rake run
```

Once the server is running go to '[https://your\\_codio\\_name-4567.codio.io/](https://your_codio_name-4567.codio.io/)'. Where 'your\_codio\_name' is the name of your specific codio domain.

## Walkthrough the app

To help you see all of our features we've outlined our website by page and talked about all the features of each page.

Firstly, as an overview this is our site map for our web-application:



## Welcome page

Upon running the server and loading the web app you'll be greeted by our welcome page, which is available to view by anyone.

*Note: The 'login' link can be found on the top right of the navigation bar and the 'logout' link and 'my account' link can be found at the bottom of the footer.*

## Login

To login, first click the 'login' link on the right of the navigation bar. This will redirect you to Twitter to login.

After logging into Twitter it will redirect you back to the welcome page but this time instead of seeing 'login' on the right of the navigation bar you should see 'my account' or 'admin dashboard' depending on which account you used to log in.

*Note: To access the admin dashboard and functionalities you **must** use the details for the ise19team29 or altataxis Twitter account as given above.*

## My account

To go to 'my account' click the link in the top right of the navigation bar (if logged in and not an admin) or scroll to the footer of the welcome page and click the 'my account' link there.

Features of the 'my account' page:

- A summary section
- Delete account button
- A history section where users can view all their previous rides
  - All journey's view which allows users to rate and comment about each individual journey
  - Search view which allows users to easily view and filter all journeys
- A general feedback and rating section

*Note: To go back to the home page or logout click the respective links at the footer of the page or click the links in the header (logo or logout picture).*

## Admin dashboard

To access the admin dashboard you **need** to login as an admin (ise19team29 or altataxis).

Firstly the dashboard page is split into 3 sections:

1. Incoming tweets
2. Add a journey to the database section
3. List of available and unavailable taxis

The idea behind this layout is to see all tweets incoming and be able to reply and then add a journey by taking details from the tweets and assigning a taxi to that journey.

Four buttons on the incoming tweets section:

1. Tick button: allows the admin to reply to the tweets and sets info in the journey section
2. X button: removes tweet from display as it is not relevant anymore
3. i button: which sends the admin to the link of the incoming tweet
4. Garbage button: deletes the tweet from timeline of ise19team29 (only shows up on replies made by ise19team29 to users, in case they have made an error)

With this layout it's very easy for the employee to reply and assign taxis to the users. It also means the information is easily and quickly added to the database so the employee can assign more taxis to different users.

## Analytics

In the analytics page admins can view statistics on:

- Total number of users
- Total number of journeys
- Total amount of feedback
- Number of taxis registered

Each include the current total as well as a detailed graph into how they have changed over each day, giving the employees the chance to see their ratings, growth and profits.

## Twitter marketing

The Twitter marketing page allows the employees to send tweets and follow users through keywords.

Simply type a tweet message and press the tweet button. Or enter a keyword and press the follow button.

## Admin settings

The first feature in the admin settings is the way to change the current deal on offer for number of free rides. Simply enter a different number and press edit or the enter key.

The rest of the settings page is split into tables for the:

- User data
- Journey data
- Taxi data
- Feedback data

For each you can view all the data stored in the database and on top of that manually edit, delete or add to each of the tables.

## Logout

Finally, the 'logout' link logs the user out of Twitter and redirects the user back to the welcome page where they can no longer access other features unless they login again.

## Story cards

This iteration has resulted in several changes where we removed some story cards when coming across some difficulties, as well as additional stories due to changes in the specification from our clients in this second iteration. Changes, notes and annotations are highlighted in yellow.

### Story cards removed

Story	Test verifications	Time and priorities	Notes/annotations
As an employee I want to be able to edit the welcome page so that I can keep information up to date and the customer can be informed about our service	Verify that the employee can edit welcome page text in dashboard	Time Estimation: 3  Customer Priority: Medium	This story was removed because we realised that it would be difficult to implement and that it wasn't a feature which was necessary or added much as a whole.
As an employee I want to be able to extract the location from the tweet so that a customer doesn't need to specify start location	Verify that the customer has location services turned on  Verify that the employee is able to contact customers if location services aren't turned on to get correct information by direct messaging	Time Estimation: 1  Customer Priority: High	This story was removed as Twitter location would need to be on for it to work and even when it was on the location wasn't accurate nor specific enough to get a pick-up location for a taxi.
As a customer I want to be able to have access to my own data and edit it so that I can keep my information up to date	Verify that the customer can edit their own name and nickname on the user table in the database. They will have no access to anything else (only have access to their record in database)  Verify that the customer can delete their own account	Time Estimation: 2  Customer Priority: Medium	This story was removed due to how we implemented our tables in the database which meant that users didn't have any information that they could edit. It was replaced with the story to delete your own account.

## First iteration story cards

Story	Test verifications	Time and priorities	Notes/annotations
As an employee I want to be able to see all incoming tweets in the dashboard so that I can categorise them as relevant or irrelevant.	<p>Verify that the employee can interact with tweets in the dashboard</p> <p>Verify that the employee can remove irrelevant tweets from dashboard</p> <p>Verify that the employee can reply on the dashboard directly</p>	<p>Time Estimation: 2</p> <p>Customer Priority: High</p>	Removed DM as an option.
As an employee I want to be able to reply or DM the customer so that I can communicate with them.	<p>Verify that the employee is able to reply through their dashboard</p> <p>Verify that the employee can gather more information if required</p> <p>Verify that the employee can send confirmation and information of the taxi being sent to the customer</p>	<p>Time Estimation: 3</p> <p>Customer Priority: High</p>	Removed DM as an option.
As an employee I want to be able to have a journey table in the database so that I can track all journeys.	Verify that the journey information will be added to the database	<p>Time Estimation: 4</p> <p>Customer Priority: Medium</p>	Removed field requirements as changes were made to the database structure.
As an employee I want to be able to have a complaints table in the database so that I can view company's issues.	Verify that any complaint figure will be added to this database	<p>Time Estimation: 4</p> <p>Customer Priority: Low</p>	Removed field requirements as changes were made to the database structure.
As an employee I want to be able to have a customer table in the database so that I can track all customers.	Verify that the every new customer's information will be added to this database	<p>Time Estimation: 4</p> <p>Customer Priority: Medium</p>	Removed field requirements as changes were made to the database structure.



As an employee I want to be able to have a taxi table in the database so that I can track all taxis.	Verify that the employee can add taxi information into the taxi database	Time Estimation: 4  Customer Priority: Medium	Removed field requirements as changes were made to the database structure.
As an employee I want to be able to follow people who are tweeting specific keywords so that I can advertise our service effectively.	Verify that the employee can set the keywords searched for  Verify that the system can automatically follow people who follow us and tweet at us	Time Estimation: 3  Customer Priority: Medium	

## Second iteration story cards

### Story cards changed from first iteration

Story	Test verifications	Time and priorities	Notes/annotations	Assigned to
As a customer I want to be able to sign up to the website so that I can get access to deals and promotional offers.	Verify that the customer can sign up on website and then log in  Verify that the customer can delete their account  Verify that the customer can sign up through Twitter	Time Estimation: 2  Customer Priority: High	Removed verification that "the customer can edit additional information (Full Name and nickname)" as that story and requirement was removed.	Tom Tonner,  Kassandra Yuen
As a customer I want to be able to send a tweet so that I can get a taxi.	Verify that the customer has provided a valid type of taxi  Verify the customer receives DMs from the employee to gather information if required info wasn't specified in the original tweet  Verify the customer receives a reply with	Time Estimation: 5  Customer Priority: High	Removed "Verify that the customer has provided a valid address" as this was out of scope.  Also removed "Verify that the employee can extract the tweet's geolocation if the start location was	Manas Sarpatwar

	information and confirmation of the taxi being sent (Reg plate, contact number, journey ID etc.)		not specified" as this is no longer a story and a requirement.  Changed DM to reply.	
As a customer I want to be able to specify a different start location so that I can order a taxi from anywhere.	Verify that the employee can contact customer if the address is unknown, is not specific enough or is unspecified	Time Estimation: 1  Customer Priority: High	"Verify that the customer has tweeted a valid Sheffield address" was out of scope.	Manas Sarpatwar
As a customer I want to be able to see the history of my bookings so that I can view how my journeys and view my offers.	Verify that the customer is signed up with the service to count towards any offers  Verify that the all journey information will be viewed in a dashboard (Taxi Reg, Time, Start/End Location, etc.)	Time Estimation: 2  Customer Priority: Medium	"Verify that the customer can see all journeys that they weren't signed up for (marked as not counting towards offers)" was removed as it's out of scope and unnecessary.	Kassandra Yuen, Chan Ken Lok
As a customer I want to be able to cancel my journey if necessary so that I have the choice to change my mind.	Verify that the employee will receive the cancellation message in a tweet so that they can cancel the order on their dashboard  Verify that the confirmation of cancellation will be received by the customer	Time Estimation: 2  Customer Priority: Medium	Removed "Verify that the customer cannot cancel the journey if the journey is marked as completed".	Tom Tonner
As a customer I want to be able to view ratings of the service so that I can make an informed decision to use service or not.	Verify that the customer can view average star ratings of the service.	Time Estimation: 1  Customer Priority: Low	"Verify that the customer can view analytics based on weekly, monthly, and yearly statistics of customer reviews" was removed as it was out of scope.	Kassandra Yuen, Manas Sarpatwar

As a customer I want to be able to rate each journey on a star system so that employees can receive feedback.	<p>Verify that the other customer can view ratings</p> <p>Verify that the customer can rate through Twitter if not signed up (employee manual adds the rating into the database)</p>	<p>Time Estimation: 3</p> <p>Customer Priority: Low</p>	<p>"Verify that the customer has signed up to the website if using contact form on the website" didn't apply after implementing the system.</p> <p>"Verify that the customer can see rating on history dashboard " was out of scope.</p>	Chan Ken Lok, Manas Sarpatwar
As a customer I want to be able to make a complaint so that I can express feelings towards the service and suggest change.	<p>Verify that the customer can complain via a contact form on the website</p> <p>Verify that the customer's complaints are recorded and marked</p> <p>Verify that the customer cannot view private complaints</p>	<p>Time Estimation: 2</p> <p>Customer Priority: Low</p>	<p>"Verify that the customer has completed a journey to file a complaint" is an unnecessary requirement.</p>	Chan Ken Lok, Manas Sarpatwar
As an employee I want to be able to see analytics of our service so that we can adjust our business strategy.	Verify that the employee can see data and graphs on number of journeys, ratings of journeys, followers, number of complaints etc.	<p>Time Estimation: 4</p> <p>Customer Priority: Medium</p>	<p>Removed "Verify that the employee can change to weekly, monthly, yearly statistics" as it was out of scope.</p> <p>Changed the data that will be seen by the employee.</p>	Simonas Petkevicius

### Unchanged story cards tackled in this iteration

Story	Test verifications	Time and priorities	Assigned to
As a customer I want to be able to choose a taxi type when I order so that it can meet my requirements.	<p>Verify that the customer can choose the type of taxi</p> <p>Verify that the customer has chosen a valid type of taxi</p>	<p>Time Estimation: 2</p> <p>Customer Priority: High</p>	Tom Tonner, Chan Ken Lok

	Verify that the customer's chosen type of taxi is currently available, if not then employee can contact the customer to ask them to wait or allocate them a different type of taxi		
As an employee I want to be able to manage offers and discounts so that I can give frequent customers benefits.	<p>Verify that the employee can change every n-th ride granting a customer a free ride (customers that are signed up on the website only)</p> <p>Give any account free rides/offers manually</p>	<p>Time Estimation: 2</p> <p>Customer Priority: High</p>	<p>Manas Sarpatwar,</p> <p>Tom Tonner</p>
As an employee I want to be able to manage offers and discounts so that I can give frequent customers benefits.	<p>Verify that the employee can change every n-th ride granting a customer a free ride (customers that are signed up on the website only)</p> <p>Give any account free rides/offers manually</p>	<p>Time Estimation: 2</p> <p>Customer Priority: High</p>	<p>Manas Sarpatwar,</p> <p>Tom Tonner</p>
As an employee I want to be able to assign taxis to customers so that I can allocate and send taxi information to the customer.	<p>Verify that the employee can contact the customer for any issues</p> <p>Verify that the employee has the option to cancel any journey and thus make the taxi assigned available again</p> <p>Verify that the system marks taxis unavailable after allocation</p> <p>Verify that the system adds the relevant information into the journey database</p>	<p>Time Estimation: 5</p> <p>Customer Priority: High</p>	Manas Sarpatwar
As an employee I want to be able to have access to all databases so that I can edit any information.	<p>Verify that the employee has access to edit any and all information on Journey, Taxi, Complaint and Customer databases</p> <p>Verify that the employee can add/remove/edit all records in all databases</p>	<p>Time Estimation: 4</p> <p>Customer Priority: Medium</p>	<p>Tom Tonner</p> <p>Chan Ken Lok</p>

	Verify that the employee can view and edit all records in dashboard		
--	---	--	--

### Story cards added from first iteration

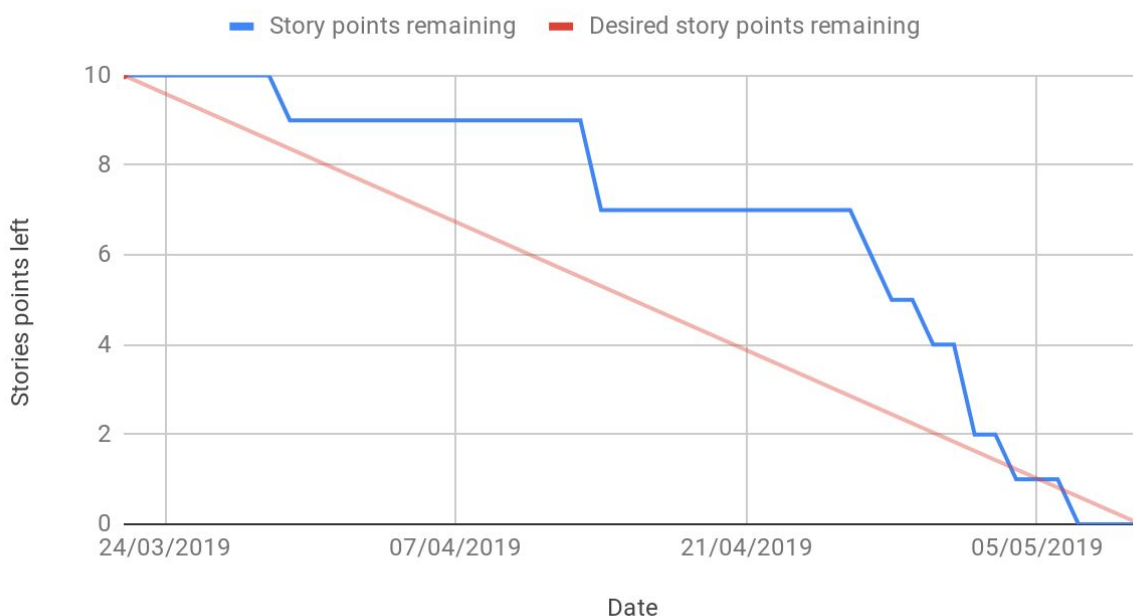
Story	Test verifications	Time and priorities	Notes/annotations	Assigned to
As a customer I want to be able to get a taxi from Manchester as well as Sheffield.	<p>Verify that the customer can specify from Manchester or Sheffield.</p> <p>Verify that the employee can assign Manchester/Sheffield only taxis depending on the employee's location</p> <p>Verify that the customer can get a taxi in both cities</p>	<p>Time Estimation: 1</p> <p>Customer Priority: High</p>	<p>This was the new requirement given by our clients after showing them the first iteration.</p> <p>This was fairly easy to implement as all we had to add was another field in the taxi table for 'city' and have another user type in the user table.</p>	Manas Sarpatwar
As a customer I want to be able to delete my account from your service if I wish.	Verify that the customer can delete their own account from the user table in the database and they will have no access to anything except their record in the database	<p>Time Estimation: 2</p> <p>Customer Priority: Medium</p>	This, as discussed earlier, was replaced by the customer edit story as it wasn't viable to implement it as it was in earlier iterations.	Chan Ken Lok
As an employee I should be able to view the available and unavailable taxis for the city I am in	<p>Verify employee sees only the taxis for their city.</p> <p>Verify system works for assigning taxis in both Sheffield and Manchester.</p>	<p>Time Estimation: 1</p> <p>Customer Priority: High</p>	<p>This was the new requirement given by our clients after showing them the first iteration.</p> <p>This just used the table changes stated earlier to filter the taxis show to the employee.</p>	Manas Sarpatwar
As an employee I should be able to respond to tweets	Verify employee can respond to tweets in both cities.	<p>Time Estimation: 1</p>	This was the new requirement given by our clients after	N/A

from Sheffield or Manchester depending on the city I am in	Verify employee sees all tweets but can mark other city tweets as irrelevant.	Customer Priority: High	<p>showing them the first iteration.</p> <p>Seeing tweets only coming from the city wasn't implemented as many people don't have location services on and it seemed an unnecessary task when the employee can mark it as irrelevant.</p>	
--	---	-------------------------	--	--

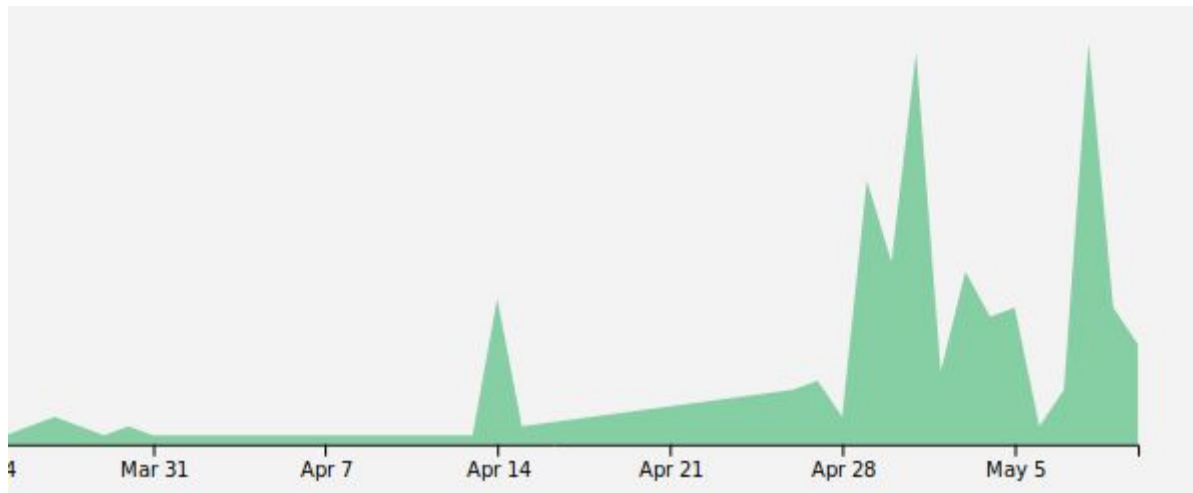
## Story card burndown chart

Our story card burndown chart for our second iteration:

Story burndown chart for iteration 2



And this is the commit contributions graph supplied by GitLab:



The graphs show that we did most of our work as a team towards the end. This is mainly because the easter break was at the start of this iteration and we all had multiple other assignments to complete during that time.

Once the easter break had ended, commits and stories were completed very rapidly as everyone could spend a lot of time on completing their tasks.

Obviously, the burndown chart for this iteration isn't ideal, but we still feel that once we started proper sprints after Easter everyone worked together well and efficiently, meaning we had the stories complete well within the given time.

## Testing

For testing in this iteration we completed both cucumber and minitests, as well as lots of manual testing to ensure our program was working as intended.

### Manual tests

Assigned to- Ibrahim Sinan Bicer, Su Zhong Ying.

To start with we ran and recorded manual tests as they were easier to run, test and edit during development. Thus, we recorded the manual tests using the following table for readability:

Test	Test for	Expected outcome	Actual outcome	Pass?	Modifications
1	Main Dashboard/ Adding a new journey	Adds a new journey to database for valid inputs, and error message for non-valid inputs	Journeys added successfully with valid inputs. Journey gave error when non-valid input used	Pass	N/A

2	Settings Section-> Editing journeys/Taxi ID	Test with valid inputs (Numbers)	Taxi ID edited successfully	Pass	N/A
3	Settings Section-> Editing journeys/Taxi ID	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...)	!!Taxi ID edited successfully!! (Taxi ID must be a positive number)	Fail	Inputs must only focus on positive numbers
4	Settings Section-> Editing journeys/User ID	Test with valid inputs (Anything but Null)	UserID edited successfully	Pass	N/A
5	Settings Section-> Editing journeys/User ID	Test with non-valid inputs (Null)	UserID gives error	Pass	N/A
6	Settings Section-> Editing journeys/Twitter Handler	Test with valid inputs (Anything but Null)	Twitter Handle edited successfully	Pass	N/A
7	Settings Section-> Editing journeys/Twitter Handler	Test with non-valid inputs (Null)	Twitter Handle gives error	Pass	N/A
8	Settings Section-> Editing journeys/Date Time	Test with valid inputs (Date with numbers)	Date time edited successfully	Pass	N/A
9	Settings Section-> Editing journeys/Date Time	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...)	!! Date time edited successfully!! (Date time must include positive numbers)	Fail	Inputs must only focus on reasonable dates with numbers
10	Settings Section-> Editing journeys/Start-En d Location	Test with valid inputs (Place Names "Strings")	Start- End Location edited successfully	Pass	N/A
11	Settings Section-> Editing journeys/Start-En d Location	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...))	!! Start- End Location edited successfully!! (Start-End location must be a place name)	Fail	Inputs can focus on reasonable place names
12	Settings Section-> Editing journeys/Free Ride	Test with valid inputs (0 and 1)	Free Ride edited successfully	Pass	N/A
13	Settings Section-> Editing journeys/Free Ride	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...))	Free Ride gives error	Pass	N/A
14	Settings Section-> Editing	Test with valid inputs (0 and 1)	Cancelled Rides edited successfully	Pass	N/A



	journeys/Cancelled Rides				
15	Settings Section-> Editing journeys/Cancelled Rides	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...))	Cancelled Rides gives error	Pass	N/A
16	Settings Section-> Editing journeys/Rating	Test with valid inputs (0, 1, 2, 3, 4, 5)	Rating edited successfully	Pass	N/A
17	Settings Section-> Editing journeys/Rating	Test with non-valid inputs (Punctuations, Emojis, Names, Negative Numbers...))	Rating Rides gives error	Pass	N/A
18	Main Dashboard -> Available Taxis/Unavailable Taxis	Available to unavailable/ Unavailable to available	Taxis in available section has taken to unavailable section during their journey/ Reversed situation for unavailable taxis section	Pass	N/A
19	Marketing Section/ Following people with a keyword	5 people (Max limit of Twitter) should be followed by a keyword	5 people followed by using a keyword	Pass	N/A
20	Marketing Section/ Sending Tweet	A tweet should be sent by marketing section	Tweet has been sent	Pass	N/A
21	Settings Section/ User table	Adding new user / Editing a user/ Deleting a user	New user added/edited/deleted successfully	Pass	N/A
22	Settings Section/ Taxi table	Adding new taxi / Editing a taxi / Deleting a taxi	New taxi added/edited/deleted successfully	Pass	N/A
23	Settings Section/ Feedback table	Adding new feedback / Editing a feedback / Deleting a feedback	New feedback added/edited/deleted successfully	Pass	N/A
24	Dashboard page/ Reply to tweet, destroy tweet: Simulating 200 times concurrently	Rate limit exceeded error should not be raised	User status update limit exceeded error	Pass	N/A
25	My account page/ Delete account	Successfully delete the account and sign out the user	Admin user deleted and when signed in again, made a non admin by default as a new user.	Pass	N/A

## The automated tests

After most of the main development was done we went back and developed automated tests to make sure our system was fully tested and testing was rigorous.

However, in order for cucumber tests to run on codio, we set up 'selenium's chrome headless javascript driver' for capybara which allows us to sign in to Twitter and validate ajax content in the tests. In order to run this driver, an external chromedriver has to be installed on codio, which takes a fair amount of time to install. Due to a majority of the tests running on the admin dashboard, each test scenario requires the user to be logged in as admin and as such the tests tagged with '@javascript' run slowly.

If tests are run multiple times in quick succession then the login feature will fail as Twitter will think the login activity as suspicious and lock the account for a short while.

The mintests have 41 separate test cases for 6 different methods. The cucumber has 46 scenarios covering most of the rest of the code and features.

```
Run options: --seed 38440

# Running:

.....

Finished in 0.245909s, 166.7282 runs/s, 166.7282 assertions/s.

41 runs, 41 assertions, 0 failures, 0 errors, 0 skips
Coverage report generated for MiniTest to /home/codio/workspace/project/coverage. 55 / 55 LOC (100.0%) covered.
```

### *Minitest results*

```
46 scenarios (46 passed)
329 steps (329 passed)
4m5.976s
Coverage report generated for Cucumber Features, MiniTest to /home/codio/workspace/project/coverage. 565 / 593 LOC (95.28%) covered.
```

### *Cucumber, Minitest results*

All Files (95.28%)							Generated about a minute ago
All Files (95.28% covered at 8.2 hits/line)							
9 files in total. 593 relevant lines. 565 lines covered and 28 lines missed							
Search:							
File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line	
app/scheduler.rb	81.82 %	25	11	9	2	0.8	
admin/marketing.rb	82.35 %	72	34	28	6	15.1	
admin/dashboard.rb	90.18 %	172	112	101	11	2.6	
app/main.rb	92.86 %	46	28	26	2	7.0	
admin/login.rb	93.94 %	58	33	31	2	22.8	
app.rb	95.12 %	58	41	39	2	23.5	
database/editDatabases.rb	98.82 %	464	255	252	3	3.0	
admin/analytics.rb	100.0 %	38	24	24	0	1.9	
app/methods.rb	100.0 %	98	55	55	0	24.1	
Showing 1 to 9 of 9 entries							
Generated by simplecov v0.16.1 and simplecov-html v0.10.2 using Cucumber Features, MiniTest							

### *Coverage levels*

The testing combined covers 95.28% of the lines of code written. The ~5% not covered is mainly due to a rescue clause that the twitter API methods have for hitting a Twitter rate limit error. Although steps have been taken so that a user does not abuse the buttons and hit the

rate limit, a rescue clause is present in case it ever is raised. We attempted to cover this manually in the manual test #24. Other lines not covered required specific scenarios and looping over set of adding data, which would have further slowed down the testing time, so were omitted in the automated testing. A more specific scenario of deleting account of a user was tested manually in manual test #25.

To run the cucumber and minitest together, make sure you're in the project directory (using `cd` if needed) and simply enter the following command in the terminal:

```
rake test
```

To run the respective tests on their own type in the terminal either:

```
rake cucumber or rake minitests
```

The cucumber tests require a test database to be created. Before running individual feature files, run:

```
rake createtestdb
```

After running individual feature files, run:

```
rake clean
```

to delete the created test database.