

COM1001 Team Project: ISE 2019 Team 29's Report (Iteration 1) - Alta Taxis

Contents

Contents	1
Members and Teamwork	1
Twitter Info	2
System setup before running (very important)	2
Changing ruby version of codio	2
Clone our git repository	3
Setting up callback URL	3
Gem requirements	5
How to run the app	5
Running the Automated Tests	5
Installing chromedriver	5
Walkthrough the app	5
Welcome page	6
Login	7
My Account	8
Admin dashboard	8
Story cards	11
Customer Story Cards	11
Employee Story Cards	17
Stories tackled in this iteration	24
Story card burndown chart	27
Testing	28

Members and Teamwork

Below is the list of members separated into partnered groups with individual work in brackets:

Database, login and behind scenes work

-Tom Tonner [AquinasAhoy for some of the commits] (Report, databases, manual testing, general help in other areas, login)

-Chan Ken Lok (Login,databases>manual test and general help, homepage text)

Website design

-Kassandra Yuen (Homepage, User Account Page, User Order Page)

Marketing and dashboard

-Manas Sarpatwar (Dashboard, fetching tweets, displaying taxis, integrated post forms with admin dashboard using AJAX, automated testing using cucumber, error handling)

-Simonas Petkevicius (Marketing page, automated following, manual testing, a bit of dashboard, error handling)

Complaints

-Ibrahim Sinan Bicer (General manual testing and team work)

-Su Zhong Ying (General manual testing and team work)

Most of our work was done on weekends and set up meetings where every member met up and worked on the code, helping each other and communicating in person in the process.

Twitter Info

For you to have admin access and set up the application you'll need to login under our twitter account. This is the information you'll need to login:

Username: ise19team29

Password: SoftEng2019

System setup before running (very important)

Changing ruby version of codio

For our gemfile to work you'll need to run a later ruby version than that on codio. Therefore if you're using codio you'll need to update your ruby version on codio. Follow these steps to do so:

1. Install rvm on codio if not already installed <https://rvm.io/rvm/install>
2. Then:
source /home/codio/.rvm/scripts/rvm upgrade 2.6.0

Clone our git repository

Type *git clone* <https://git.shefcompsci.org.uk/com1001-2018-19/team29/project>

And enter your username and password for gitlab.

Setting up callback URL

If using localhost:4567 then ignore this step.

If using codio to run the server then follow this:

- 1) Go to <https://developer.twitter.com/content/developer-twitter/en.html>
- 2) Login as ise19team29 with the above twitter details provided
- 3) Go to <https://developer.twitter.com/en/apps>
- 4) Click the 'details' button on the 'ise19team29 - Alta' app on the list as shown:

The screenshot shows the Twitter Developer Portal interface. At the top, there's a navigation bar with links: Developer, Use cases, Products, Docs, More, Dashboard, and a dropdown for ise19team29. Below the navigation bar, the 'Apps' section is active, showing a list of three apps. Each app entry includes a Twitter logo icon, the app name, the App ID, and a 'Details' button. The apps listed are: 'ise19team29 - Alta' (App ID: 16126309), 'AutoFollowing for the app' (App ID: 16161710), and 'Marketing tab' (App ID: 16167048). At the bottom of the page, there's a footer with various links organized into categories: About, Business, Developers, Help Center, and Marketing.

App Name	App ID	Action
ise19team29 - Alta	16126309	Details
AutoFollowing for the app	16161710	Details
Marketing tab	16167048	Details

- 5) Click edit app details on the top right after clicking the edit dropdown button

Developer
 Use cases
 Products
 Docs
 More

Dashboard ise19team29

Apps

> ise19team29 - Alta

App details

Keys and tokens

Permissions

App details

Details and URLs

App icon

App icon is default, click edit to upload.

App Name

ise19team29 - Alta

Description

ise19team29's Twitter web app

Website URL

<https://www.sheffield.ac.uk/>

Sign in with Twitter

Enabled

Callback URL

<http://alex-tunnel-4567.codio.io/auth/twitter/callback>
<http://127.0.0.1:3000/auth/twitter/callback>
<http://balloon-lake-4567.codio.io/auth/twitter/callback>
<http://balloon-lake-4567.codio.io/auth/twitter>
<http://localhost:3000/auth/twitter/callback>

Edit

Edit details

Delete

- 6) Under 'Callback URL' Add a callback URL which will be your Codio link (eg. <http://motor-disney-4567.codio.io/>) with /auth/twitter/callback at the end. Full example: <http://motor-disney-4567.codio.io/auth/twitter/callback>

Developer
 Use cases
 Products
 Docs
 More

Dashboard ise19team29

<https://www.sheffield.ac.uk/>

Allow this application to be used to sign in with Twitter

Learn more

☒ Enable Sign in with Twitter

Callback URLs (required)

OAuth 1.0a applications should specify their oauth_callback URL on the request token step, which must match the URLs provided here. To restrict your application from using callbacks, leave these blank.

<http://alex-tunnel-4567.codio.io/auth/twitter/callback>

<http://balloon-lake-4567.codio.io/auth/twitter/callback>

<http://balloon-lake-4567.codio.io/auth/twitter>

<http://127.0.0.1/auth/twitter/callback>

<http://motor-disney-4567.codio.io/auth/twitter/callback>

<http://localhost:4567/auth/twitter/callback>

<http://127.0.0.1/>

+ Add another

Terms of Service URL

<https://>

Privacy policy URL

<https://>

Organization name

- 7) Save at the bottom of the screen

Gem requirements

Assuming Ruby and Bundler are installed go into the main directory and run: 'bundle install'. This will install all Gem requirements needed for using our app.

How to run the app

To start running the app on your local host or on Codio go into project directory and in the console run 'ruby app.rb'.

Once the server is running go to '<http://localhost:4567/>' if running on a local machine or 'http://your_codio_name-4567.codio.io/' if you're running from Codio.

Running the Automated Tests

In order to run the cucumber tests you'll need to run them on a local machine as they won't work on codio. Full details about why, are in the testing section.

Installing chromedriver

To run automated tests you'll need to install chromedriver. This can be done by following these steps for the different systems you might be running on your local machine:

- Windows:
<http://jonathansoma.com/lede/foundations-2018/classes/selenium/selenium-windows-install/>
- Mac:

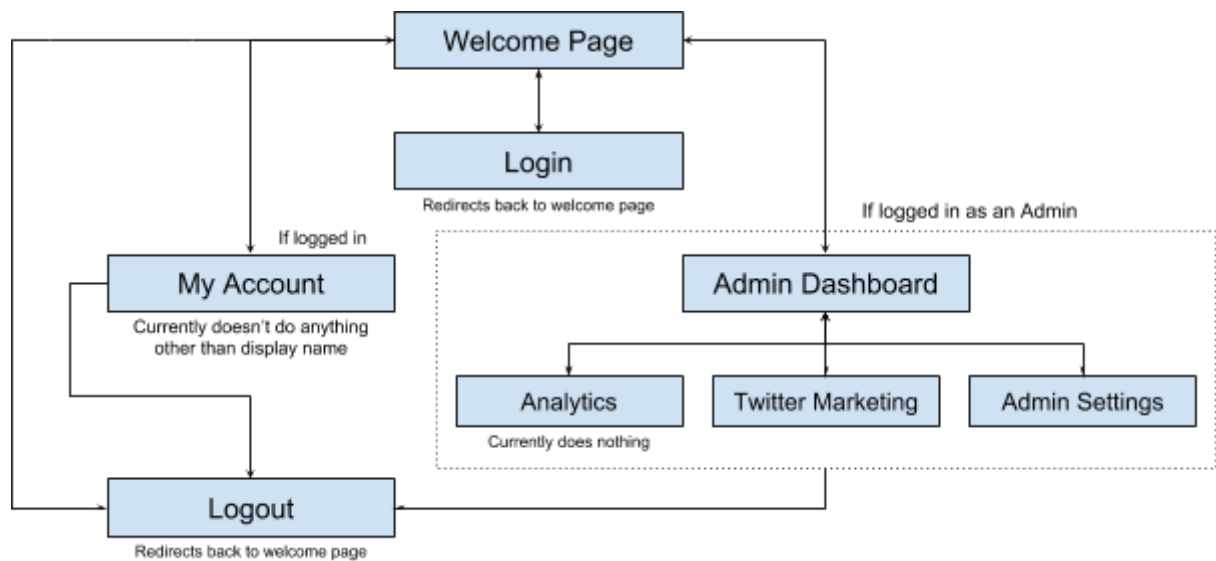
```
brew cask install chromedriver
```
- Ubuntu:

```
sudo apt-get install chromium-chromedriver
```

Then before you run the tests make sure you open app.rb and then uncomment 'session[:admin] = true' in the before filter (and comment back again if running the web app after the testing) . Then you should be able to run the tests, by running the command 'cucumber' in the project root directory.

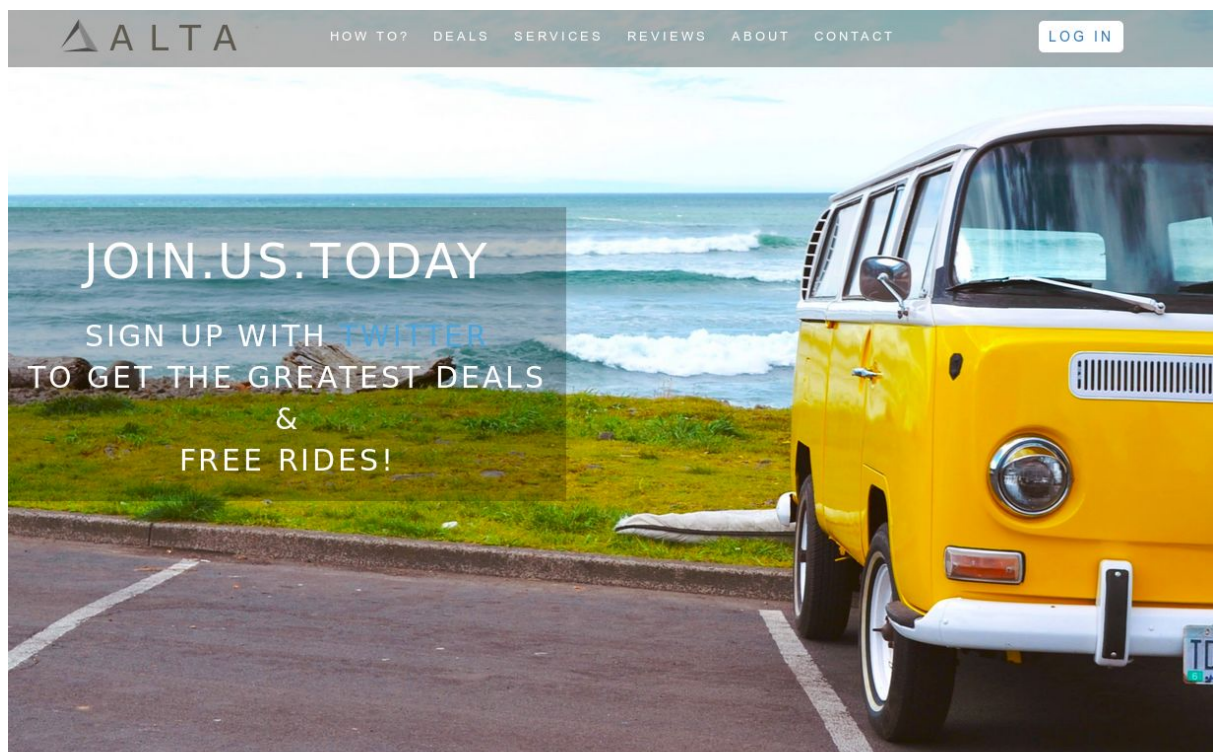
Walkthrough the app

Site map for the website in this current iteration:



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 everyone.



HOW TO USE OUR SERVICES

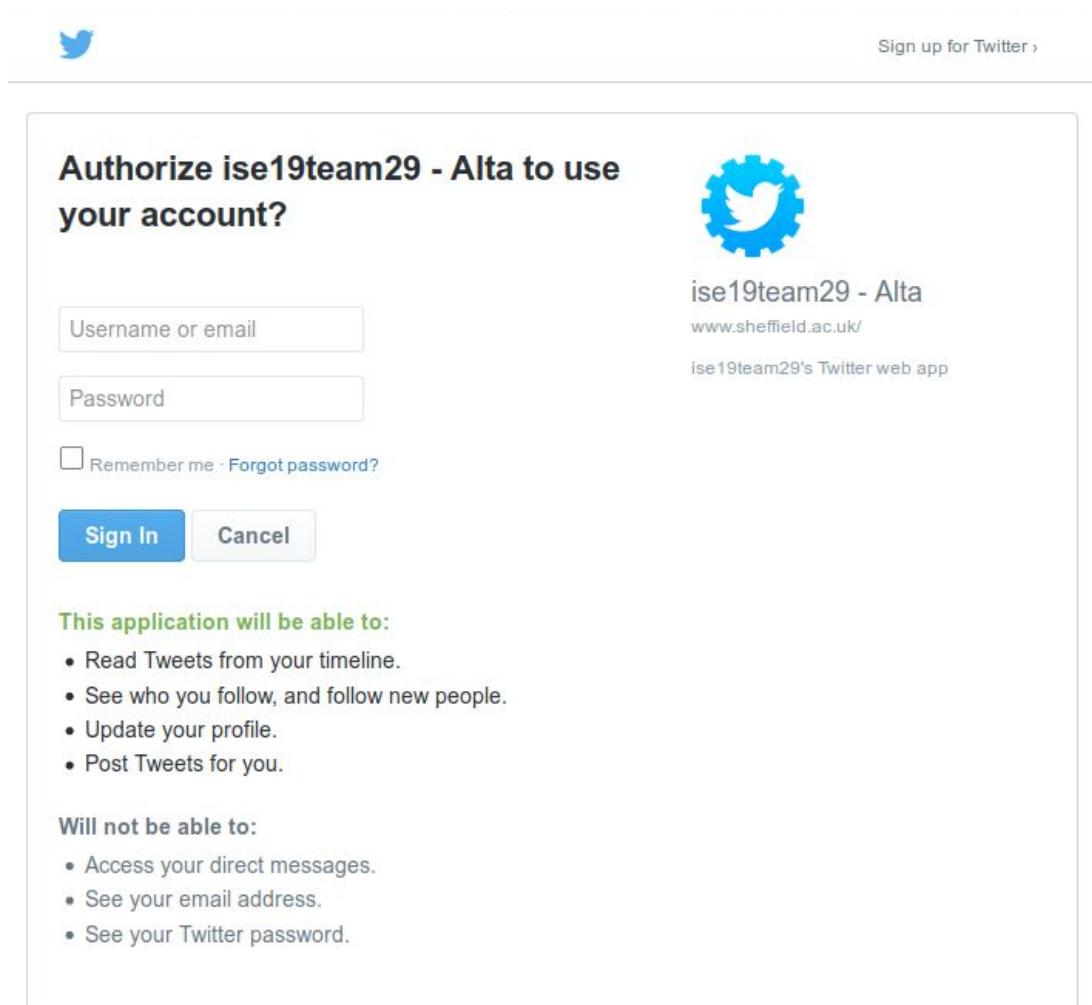
Here you'll find information on the service by scrolling down or by clicking the respective information headers in the navigation bar.

Note: 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

In order to access all other functions of the website, you'll need to click the login link.

To do this first click the login link on the right of the navigation bar. This will redirect you to Twitter to log in.



The screenshot shows the Twitter login interface. At the top left is the Twitter logo, and at the top right is a link that says "Sign up for Twitter >". The main content area is titled "Authorize ise19team29 - Alta to use your account?". To the right of this title is a blue gear icon with a white Twitter bird inside. Below the icon, the text reads "ise19team29 - Alta", "www.sheffield.ac.uk/", and "ise19team29's Twitter web app". On the left side, there are two input fields: "Username or email" and "Password". Below these is a checkbox labeled "Remember me" and a link "Forgot password?". There are two buttons: a blue "Sign In" button and a grey "Cancel" button. Below the buttons, there is a section titled "This application will be able to:" followed by a list of permissions: "Read Tweets from your timeline.", "See who you follow, and follow new people.", "Update your profile.", and "Post Tweets for you.". Below this is a section titled "Will not be able to:" followed by a list of permissions: "Access your direct messages.", "See your email address.", and "See your Twitter password."

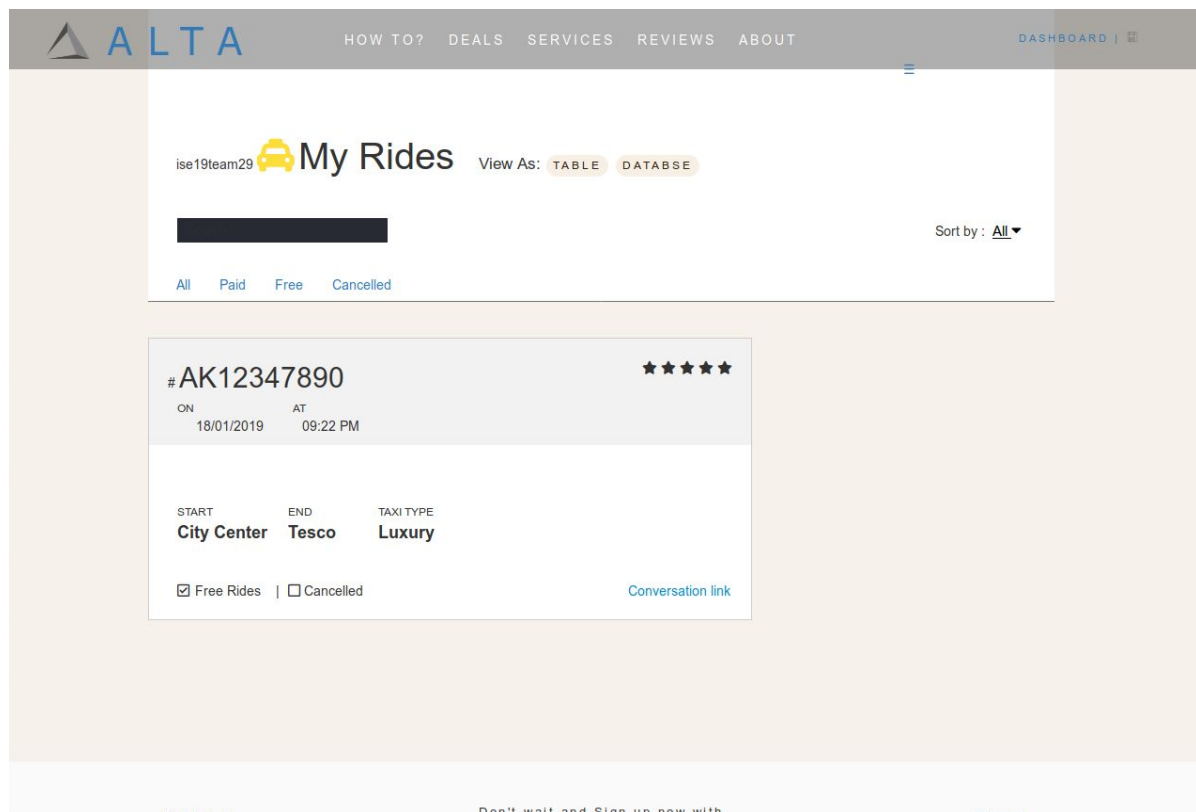
After logging into Twitter it will redirect you back to the welcome page but this time instead of seeing 'Login' on the right of the navigation bar you should see 'My account' or 'Admin Dashboard' depending on which account you used to log in.

Note: To access the admin dashboard and functionalities you MUST use the details for the ise19team29 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.

In this current iteration the my account page doesn't display useful information and is more of a prototype of what it will be in the second iteration. Currently, the name of the person logged in is displayed on the left and then by clicking orders, you can see a mock-up of what viewing journeys will be like in the final version.

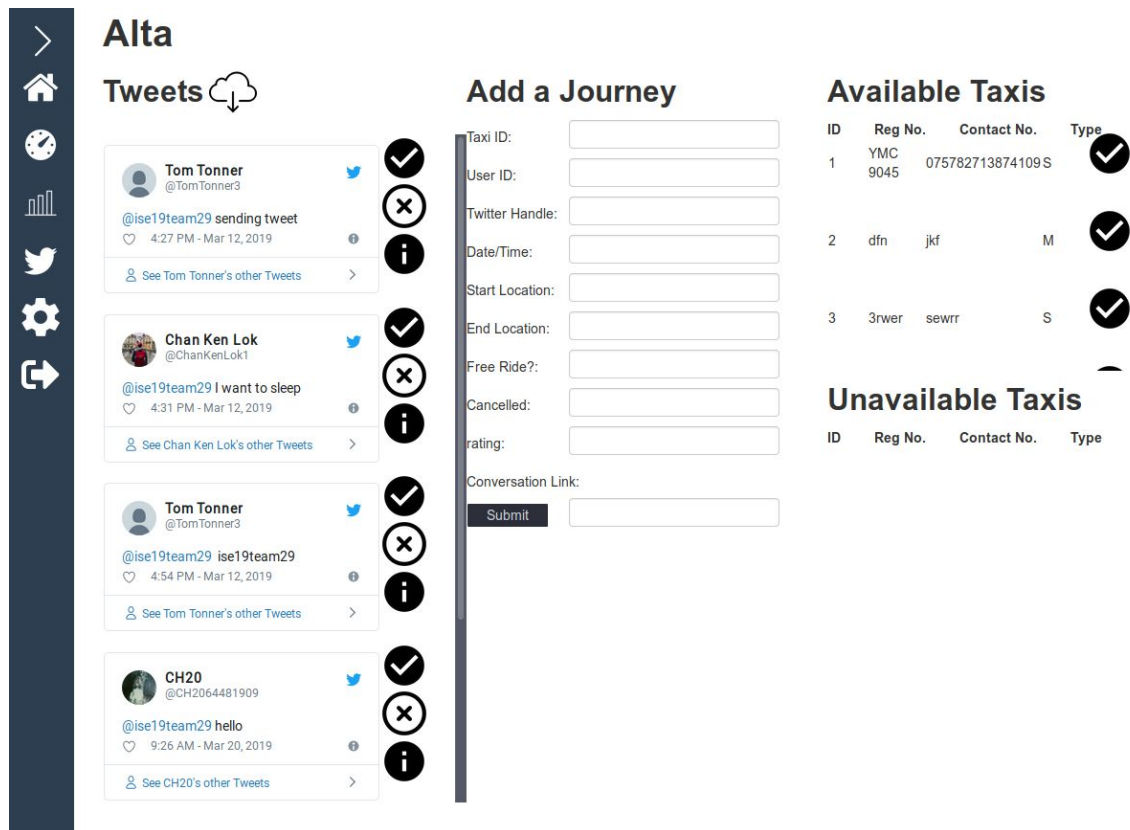


Note: To go back to the home page or logout click the respective links at the footer of the page.

Admin dashboard

To access the admin dashboard you NEED to log in as an admin (ise19team29) for which the details to log in as an admin are given in this report above (the section called: Twitter Info).

Once logged in as an admin you should see 'admin dashboard' link where the login link was on the navigation bar originally. Click this new link to direct you to the admin dashboard.



The nav bar in the dashboard now has these links:

- Home: which takes you back to the welcome page
- Dashboard: which takes you back to this main dashboard page
- Analytics: which doesn't do anything in this iteration
- Twitter: which allows the admin to custom follow and tweet people
- Settings: which allows the admin to add manually to the taxi, customer and complaints database
- Log out: which logs the admin out and redirects them to the welcome page

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

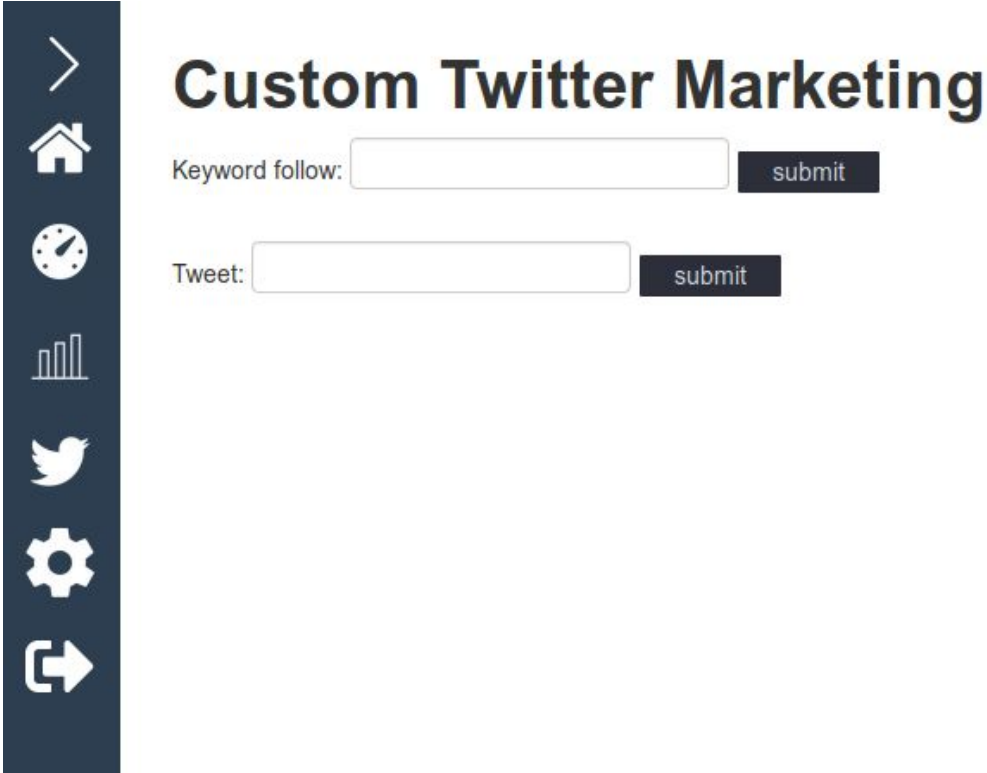
The idea behind this layout is to see all tweets incoming and be able to reply and then add a journey by taking details from the tweets and assigning a taxi to that journey.

Three buttons on the incoming tweets section:

1. Tick button: allows the admin to reply to the tweets
2. X button: removes tweet as it is not relevant anymore
3. i button: which sends the admin to the link of the tweet sent

Currently, all additions to the add journey section have to be added in manually. In the next iteration, we plan to have information added automatically when the admin assigns a taxi and when a reply is made.

The twitter marketing tab allows the admin to follow people with specific keywords used, and also send out a public tweet.



The screenshot shows a web interface for 'Custom Twitter Marketing'. On the left is a dark blue sidebar with white icons: a chevron, a house, a clock, a bar chart, a Twitter bird, a gear, and a share icon. The main content area has a title 'Custom Twitter Marketing' in bold. Below the title are two input sections. The first is labeled 'Keyword follow:' and contains a text input field followed by a dark blue 'submit' button. The second is labeled 'Tweet:' and contains a text input field followed by a dark blue 'submit' button.







Custom Twitter Marketing

Keyword follow:

Tweet:

The setting tab currently allows the admin to manually add to the other 3 tables in the database - with future iterations allowing the admin to view and edit these databases.

>



Add a User

ID:

Name:

Date/Time:

User Type:

Free Rides Left:

Submit

Add a Taxi

Reg Number:

Contact Number:

Taxi Type:

Submit

Add a Complaint

Journey ID:

User ID:

Complaint:

Submit

Finally, the logout button logs the admin out.

Story cards

This iteration has resulted in no changes to the story cards as the clients have yet to review the system and change requirements. We are also only currently partially through the story cards and haven't run into issues with them up to this point.

Customer Story Cards

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 address
- Verify that the customer has provided a valid type of taxi
- Verify that the employee can extract the tweet's geolocation if the start location was not specified
- 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 DM with information and confirmation of the taxi being sent (Reg plate, contact number, journey ID etc.)

Time Estimation: 5
Customer Priority: High

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
- Verify that the customer can edit additional information (Full Name and nickname)

Time Estimation: 2
Customer Priority: High

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 customer has tweeted a valid Sheffield address
- 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

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.)
- Verify that the customer can see all journeys that they weren't signed up for (marked as not counting towards offers)

Time Estimation: 2
Customer Priority: Medium

As a customer I want to be able to choose a taxi type when I order so that it can meet my requirements

- Verify that the customer can choose the type of taxi
- Verify that the customer has chosen a valid type of taxi
- 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

Time Estimation: 2
Customer Priority: High

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 Customer database. They will have no access to anything else (only have access to their record in database)
- Verify that the customer can delete their account
-

Time Estimation: 2
Customer Priority: Medium

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 customer cannot cancel the journey if the journey is marked as completed
- 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

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 and analytics based on weekly, monthly, and yearly statistics of customer reviews

Time Estimation: 1
Customer Priority: Low

As a customer I want to be able to rate each journey on a star system so that employees can receive feedback

- Verify that the customer has signed up to the website if using contact form on the website
- Verify that the customer can see rating on history dashboard
- Verify that the other customer can view average ratings
- Verify that the customer can rate through Twitter if not signed up (employee manual adds the rating into the database)

Time Estimation: 3
Customer Priority: Low

As an employee I want to be able to reply or DM the customer so that I can communicate with them

- Verify that the employee is able to reply/DM through their dashboard
- Verify that if the customer does not have DMs on, then the employee can send a message in a tweet to ask them to turn it on
- Verify that the employee can gather more information if required
- Verify that the employee can send confirmation and information of the taxi being sent to the customer

Time Estimation: 3
Customer Priority: High

As a customer I want to be able to make a complaint so that I can express feelings towards the service and suggest change

- Verify that the customer can complain via a contact form on the website
- Verify that the customer has completed a journey to file a complaint
- Verify that the customer's complaints are recorded and marked
- Verify that the customer cannot view private complaints

Time Estimation: 2
Customer Priority: Low

Employee Story Cards

As an employee I want to be able to manage offers and discounts so that I can give frequent customers benefits

- 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)
- Give any account free rides/offers manually

Time Estimation: 2
Customer Priority: High

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

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

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

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

- Verify that the employee can interact with tweets in the dashboard
- Verify that the employee can remove irrelevant tweets from dashboard
- Verify that the employee can reply or DM on the dashboard directly

Time Estimation: 2
Customer Priority: High

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

- Verify that the employee can contact the customer for any issues
- Verify that the employee has the option to cancel any journey and thus make the taxi assigned available again
- Verify that the system marks taxis unavailable after allocation
- Verify that the system adds the relevant information into the journey database when a taxi becomes available again

Time Estimation: 5
Customer Priority: High

As an employee I want to be able to have a journey database so that I can track all journeys

- Verify that the database has the fields: Journey ID, Customer ID (Twitter Handle), Date and time of journey, Taxi ID (Reg number), Start location, End location, Transcript, Whether it was a free ride/paid/cancelled, Journey rating
- Verify that the journey information will be added to the database

Time Estimation: 4
Customer Priority: Medium

As an employee I want to be able to have a taxi database so that I can track all taxis

- Verify that the database has the fields: Taxi ID (Reg number), Contact number, Type of taxi
- Verify that the employee can add taxi information into the taxi database

Time Estimation: 4
Customer Priority: Medium

As an employee I want to be able to have a customer database so that I can track all customers

- Verify that the database has the fields: Customer ID (Twitter Handle), Number of free rides left, Sign up date, Type of user (employee/customer)
- Verify that the every new customer's information will be added to this database

Time Estimation: 4
Customer Priority: Medium

As an employee I want to be able to have a complaints database so that I can view company's issues

- Verify that the database has the fields: Complaint text, date of complaints, Customer ID (Twitter handle) who made complaint
- Verify that any complaint figure will be added to this database

Time Estimation: 4
Customer Priority: Low

As an employee I want to be able to have access to all databases so that I can edit any information

- Verify that the employee has access to edit any and all information on Journey, Taxi, Complaint and Customer databases
- Verify that the employee can add/remove/edit all records in all databases
- Verify that the employee can view and edit all records in dashboard

Time Estimation: 4
Customer Priority: Medium

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 change to weekly, monthly, yearly statistics
- Verify that the employee can see data on tweets, retweets, number of journeys, ratings of journeys, followers, number of complaints

Time Estimation: 4
Customer Priority: Medium

Stories tackled in this iteration

The stories tackled in this iteration are listed below with our initials on the side describing who completed the story and information such as difficulty and priority also listed.

Database and tables



CL TT

Create Homepage

KY MS

As an employee I want to be able to have a journey database so that I can track all journeys



Difficulty: 4 Priority: Medium

CL SY TT

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



Difficulty: 2 Priority: High

S MS

As an employee I want to be able to have a customer database so that I can track all customers

 1   Difficulty: 4

Priority: Medium

CL TT

As an employee I want to be able to have a taxi database so that I can track all taxis


  Difficulty: 4 Priority: Medium

CL TT

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

 Difficulty: 5 Priority: High SY

As an employee I want to be able to have a complaints database so that I can view company's issues

  Difficulty: 4 Priority: Low

CL TT

Design Template

KY MS

As an employee I want to be able to reply or DM the customer so that I can communicate with them

 Difficulty: 3 Priority: High

As an employee I want to be able to follow people who are tweeting specific keywords so that I can advertise our service effectively

 Difficulty: 3 Priority: Medium



Login form/Twitter Sign in



CL TT

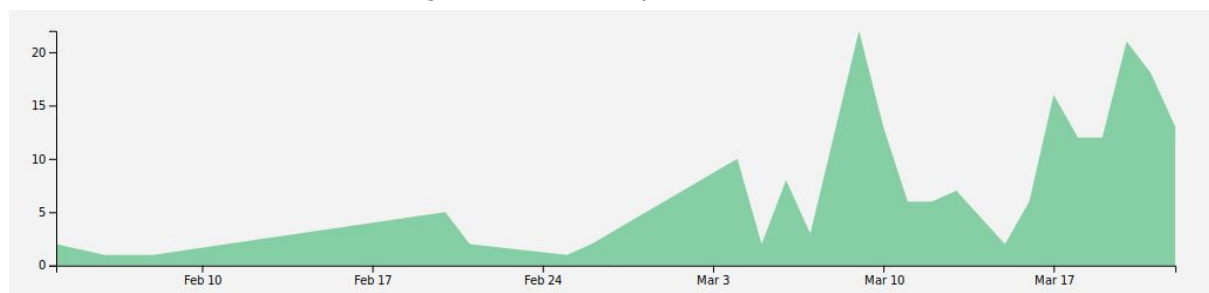
Story card burndown chart

Our story card burndown chart for our first iteration:



Note: The red line is the ideal burndown, however it isn't quite correct as we are only expected to achieve about half of the tasks by the first iteration.

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 because we spent most of our time at the start planning and learning all we could from the lectures before we made a proper start.

The large spikes of completed stories can be explained by the way in which we decided to work in which we hired out rooms to work together in person throughout the weekends.

It can also be explained by the fact that we unfortunately didn't keep our trello up-to-date all the time. This resulted in large gaps on the graph where lots of cards were completed in one go due to not keeping these cards up-to-date consistently.

Testing

For now, we are only testing the front-end behaviour with cucumber, with plans of mini-testing in the second iteration.

Unfortunately, because we are using AJAX to asynchronously add content to the page instead of normal get-post route, the errors cannot be picked up by the default driver which Cucumber uses.

Hence we had to change the default driver to Selenium, and the browser to chrome by using chromedriver. This means that Cucumber will wait for javascript content before doing the final “then” clause. Since the tests are run by firing up a browser, the tests cannot be run on codio. You will need to run the test on localhost and have the following prerequisites in-order to for the tests to pass.

- Have chrome driver installed

Installation instructions:

Windows:

<http://jonathansoma.com/lede/foundations-2018/classes/selenium/selenium-windows-install/>

Mac:

```
brew cask install chromedriver
```

Ubuntu:

```
sudo apt-get install chromium-chromedriver
```

- Make sure you open app.rb, uncomment `session[:admin] = true` in the before filter, and then run the tests, in order for Cucumber to access admin dashboard.

To run the tests make sure you are in the project directory and run `cucumber`

Following is the result of the tests. You can also find the coverage in the coverage folder.

All Files (67.03%)

Generated 27 minutes ago

All Files (67.03% covered at 1.99 hits/line)

8 files in total. 273 relevant lines. 183 lines covered and 90 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
login.rb	13.33 %	57	30	4	26	0.1
marketing.rb	30.77 %	72	39	12	27	1.8
main.rb	57.14 %	28	14	8	6	1.0
dashboard.rb	70.0 %	141	90	63	27	1.7
addTaxi.rb	93.75 %	26	16	15	1	1.8
addUser.rb	94.44 %	33	18	17	1	1.8
app.rb	96.08 %	79	51	49	2	4.2
addComplaint.rb	100.0 %	25	15	15	0	1.9

Showing 1 to 8 of 8 entries

Generated by simplecov v0.16.1 and simplecov-html v0.10.2 using Cucumber Features

```

18 scenarios (18 passed)
92 steps (92 passed)
0m28.497s
Coverage report generated for Cucumber Features to /Users/manassarpawat/Desktop/Alta/project/coverage. 183 /
273 LOC (67.03%) covered.

```

The tests cover 67.03% of the lines of code in this iteration. This is due to not being able to test login via twitter, as the automated tests get an OAuth::Unauthorized error when they are redirected to '/auth/twitter'. We did not have enough time to manually test this feature, but we will try to fix this in the next iteration.