Team names:

Satoshi Naka’know’code

Metro, Transport, Bus, Train, Uber, MetChat, RadChat, RadelaideChat, RadRoom, MetroMadness, Route-It Adelaide, BIMB

G.O.A.T – Greatest Of All Transport iiiiiiii

RadChat iiiiiiiiii

MetroMadness iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii vote for the WIN!!

RadChat - Get’n Ya There

Hey Team,

Following on from tonight’s intro and it’s great to get to know you all!

|  |
| --- |
| **Team Members** |
| Manoj Adikari |
| Islam Muhammad |
| Brea Grindley |
| Ben Armstrong |

**The Design Brief:**

***Who is the audience*?**

People who use and rely on public transport for either work/recreation to get to their chosen destination.

***What is the problems that the product solves*?**

Users can use the chat feature to talk to other commuters on the route and find out where their transport is, if it’s on time, or, has been delayed due to an accident or road works etc. This gives the end user the opportunity to find alternative transport to get to their destination on time.

***How does the product solve that problem*?**

Real live time data of where your current transport is, including if it’s on time, or late warnings. The open chat forum allows the users to connect to other users of the same route to be able to communicate freely and update each other.

**Project Title**:

API’s to Use:

* Socket.io
* Adelaide Metro API

**The End User (User Persona and/or user story):**

*Example: As a User… I want… So that…*

As a user I want to know if my transport is running on time. So that if the bus or train is not running on time or delayed, I can organise alternative transport to get to my destination.

As a user I want to create an account and have my data stored privately and securely.

Once logged into the application I want to be able to view available bus/routes from a drop down box so that I may select my appropriate route.

Once my route is selected I want to be able to get real time data of that route.

I also want to be able to connect and chat freely with other commuters on my transport route.

**It’s Done When:**

*Example: Given… When… Then…*

* Use Node.js and Express.js to create a RESTful API.
* Use Handlebars.js as the template engine.
* Use MySQL and the Sequelize ORM for the database.
* Have both GET and POST routes for retrieving and adding new data.

To satisfy Node.js, Express.js and RESTful API we will use the Adelaide Metro API

chat room for each specific bus route.

* Use at least one new library, package, or technology that we haven’t discussed.

Socket.io will satisfy new technology criteria. Socket.io will allow a real time chat feature within the application for users to communicate to each other.

* Have a folder structure that meets the MVC paradigm.

Model – Includes Data base (SQL) management and manipulation.

Viewer – We will use Handlebars.js and HTML element of Chat.

Control – API’s including homeRoutes etc.

* Include authentication (express-session and cookies).
* Protect API keys and sensitive information with environment variables.

Authentication and API Keys; All user details captured will be stored in the ‘.env’ file and ‘gitignore’ file will ensure SQL passwords will not be public.

* Be deployed using Heroku (with data).
* Have a polished UI.
* Be responsive.

CSS to have Media Queries (flex points) built into design. These should be for Tables @ 920px and Mobile Phones @ 480px

* Be interactive (in other words, accept and respond to user input).

As a user I want to be able to sign up and have my password store securely, I also want to stay logged and not asked for my credentials again until I login in the future.

As a user I want to be able to select my bus/train routes and also modify if need be.

As a user I want to use the chat feature for my chosen route to talk to other commuters.

* Meet good-quality coding standards (file structure, naming conventions, follows best practices for class/id naming conventions, indentation, quality comments, and so on).
* Have a professional README (with unique name, description, technologies used, screenshot, and link to deployed application).

**Goals (Must be Specific and Measurable):**

* Project Day 1 – Tuesday 21/03/2023: Idea decided on. Features, user persona and specific details decided on. Wire Frame to be completed.
* Project Day 2 - Thursday 23/03/2023:
* Project Day 3 – Monday 27/03/2023:
* Project Day 4 – Tuesday 28/03/2023:
* Project Day 5 – Thursday 30/03/2023:
* Project Delivery Day – Monday 03/04/2023:

**12 BUILD to Days**

Who will talk about which aspect? Noting we all need to talk at some stage.

**Who will be responsible for what**:

**Still to Be Built into the Application:**

* ~~User to be auto logged out after 5 mins of inactivity~~
* ~~DataBase~~
* ~~Routes to be captured \*can be done via route\_id~~
* ~~Log in feature~~
* ~~Sign in feature~~
* ~~Chat feature intergrated~~
* Option to disconnect from a Chat (button) \***Manoj**
* ~~Log Out Button~~
* Pull in Bus Route Information via SQL to underneath the chat name \***Manoj**
  + E.g. “503 Tea Tree Plaza via O-Bahn Busway and Holden Hill”
* ~~Add Tool Tip; feature when you hover over the map pin it will display the bus number, allowing the user to see and make sure it’s the bus they require before clicking onto it.~~
* CSS Styling \***Brea**
* Responsiveness, including “@Media Queries >> 920px for Tablets, 480px for Mobiles” \***Brea**
* Deployed and tested on Heroku \***team to get together and test/debug**
* Using fetchChatlogs.js have a feature/button to allow users to view all their previous chat history. \***Islam**
* Delete chat log/history option. \***Islam**
* Slide Deck for Presentation to be created \***Ben**

**For Future Development**:

* Using fetchChatlogs.js have a feature/button to allow users to view all their previous chat history. \***Delete if Islam gets it finished**
* Delete chat log/history option. \***Delete if Islam gets it finished**
* Add RSS feed to inform users about any delays, breakdowns and accidents.

**User Journey - NEW**

First time user Signs Up, then go’s to a second screen where I am welcomed and asked to input my bus number. (or can select a bus from the map)

The bus number will be then takes them to the ‘bus route’ chat room to be instantly connected with like minded commuters. Also on same screen the “map” pins the user to the map.

The route will be captured against the user that has signed up and stored into the data base ready for next time the user logs in.

**User Journey - EXISTING**

Existing user Logs back In then proceeds to the second screen where again I am welcomed and asked to input my bus number or most recent route displayed. (or, again, can select a different bus from the map)

The user can either select the most recent route or enter in a new bus number to take them to specific chat room.

Again on same screen the “map” pins the user to the map.

The route they select from this interaction will be captured against the user so that it will be displayed next time they log in.