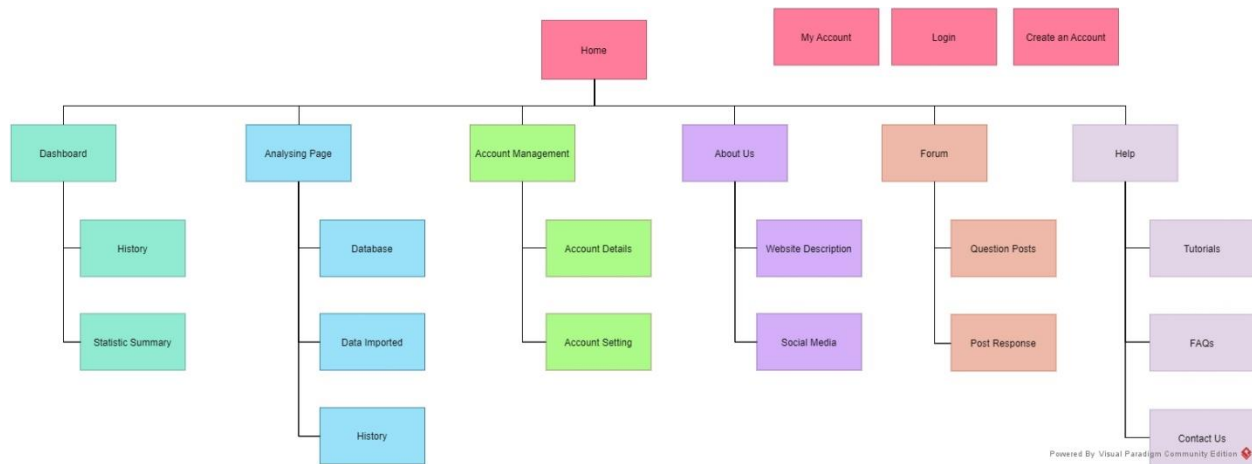


Design

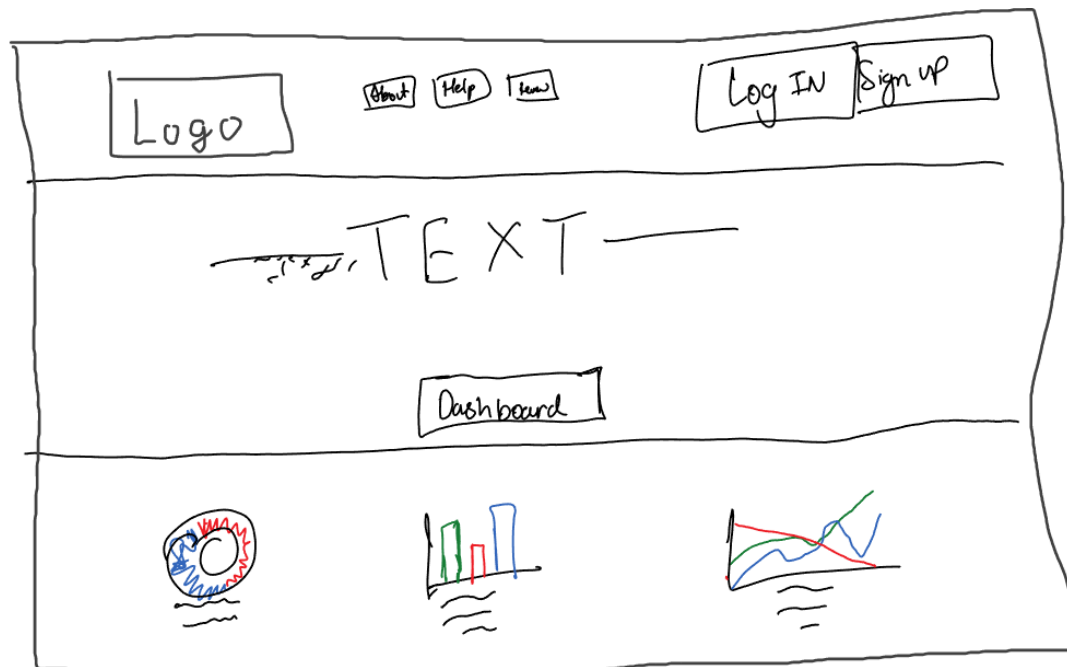
Website structure site map

A website structure site map is created to aid interface design development. The diagram maps out the structure of pages and illustrates the components under each of the pages. With the site map created in advance, the design of wireframes is more likely to fulfil all the requirements and pages are correctly connected to each other.

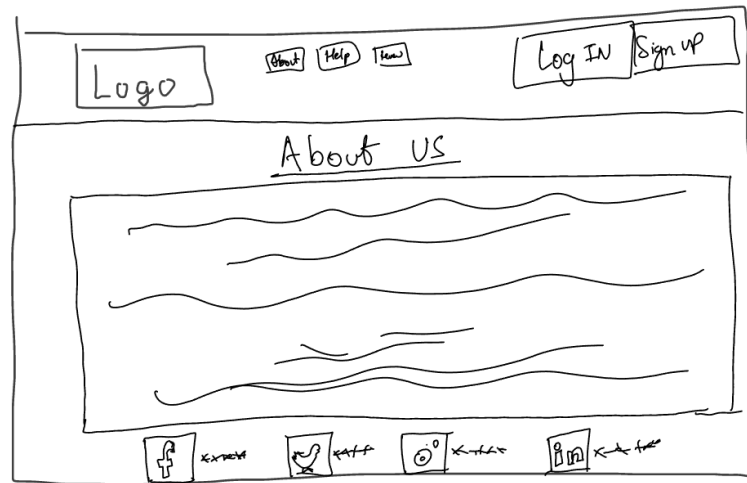


Wireframes (interface design)

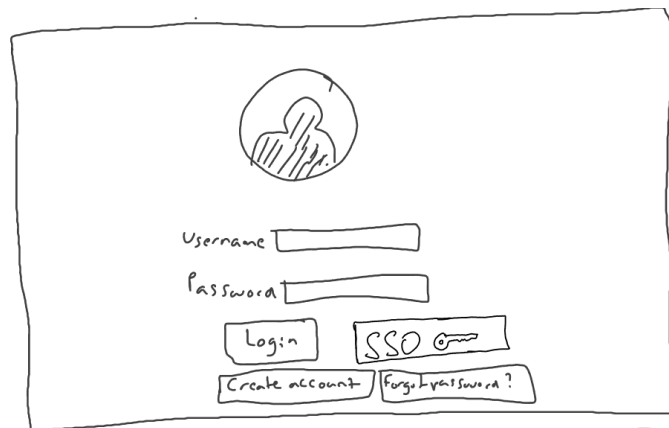
Home page:



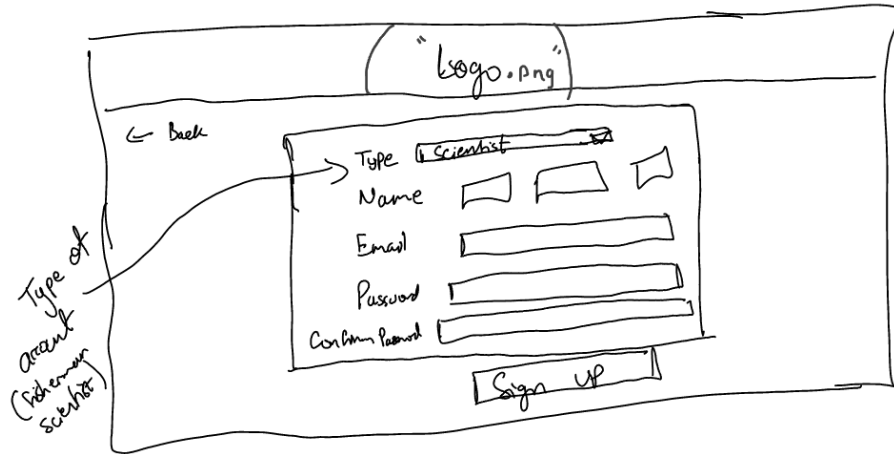
About page:



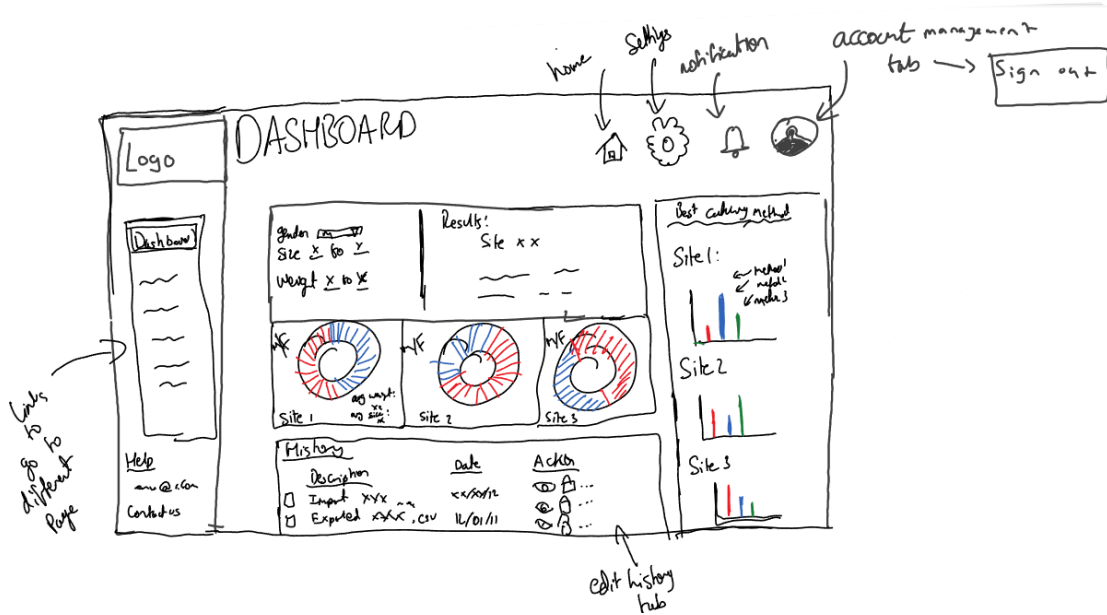
Login page:



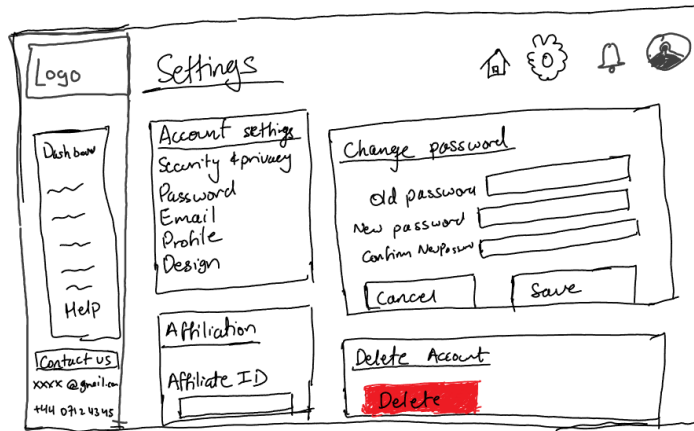
Sign-up page:



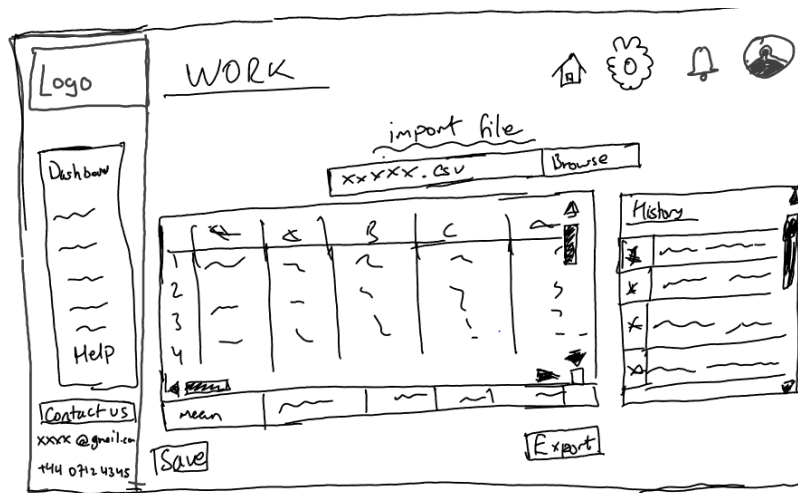
Dashboard page:



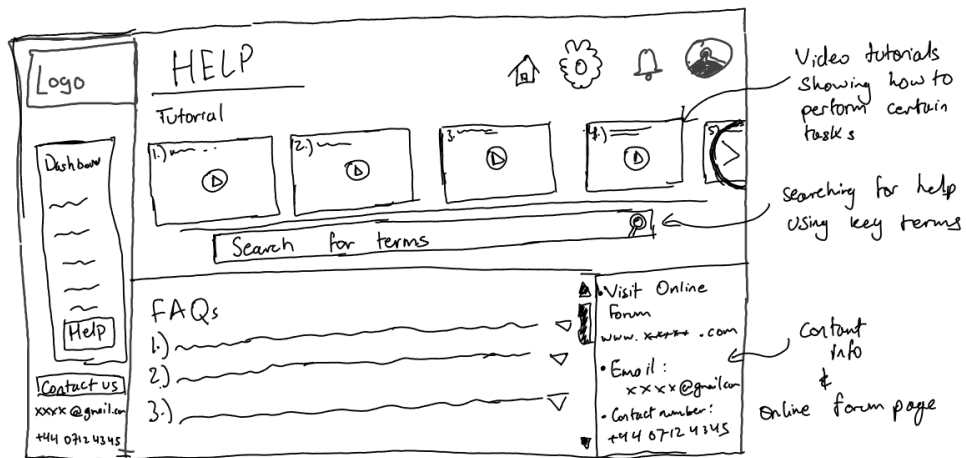
Account management page:



User analysis page:



Help page:



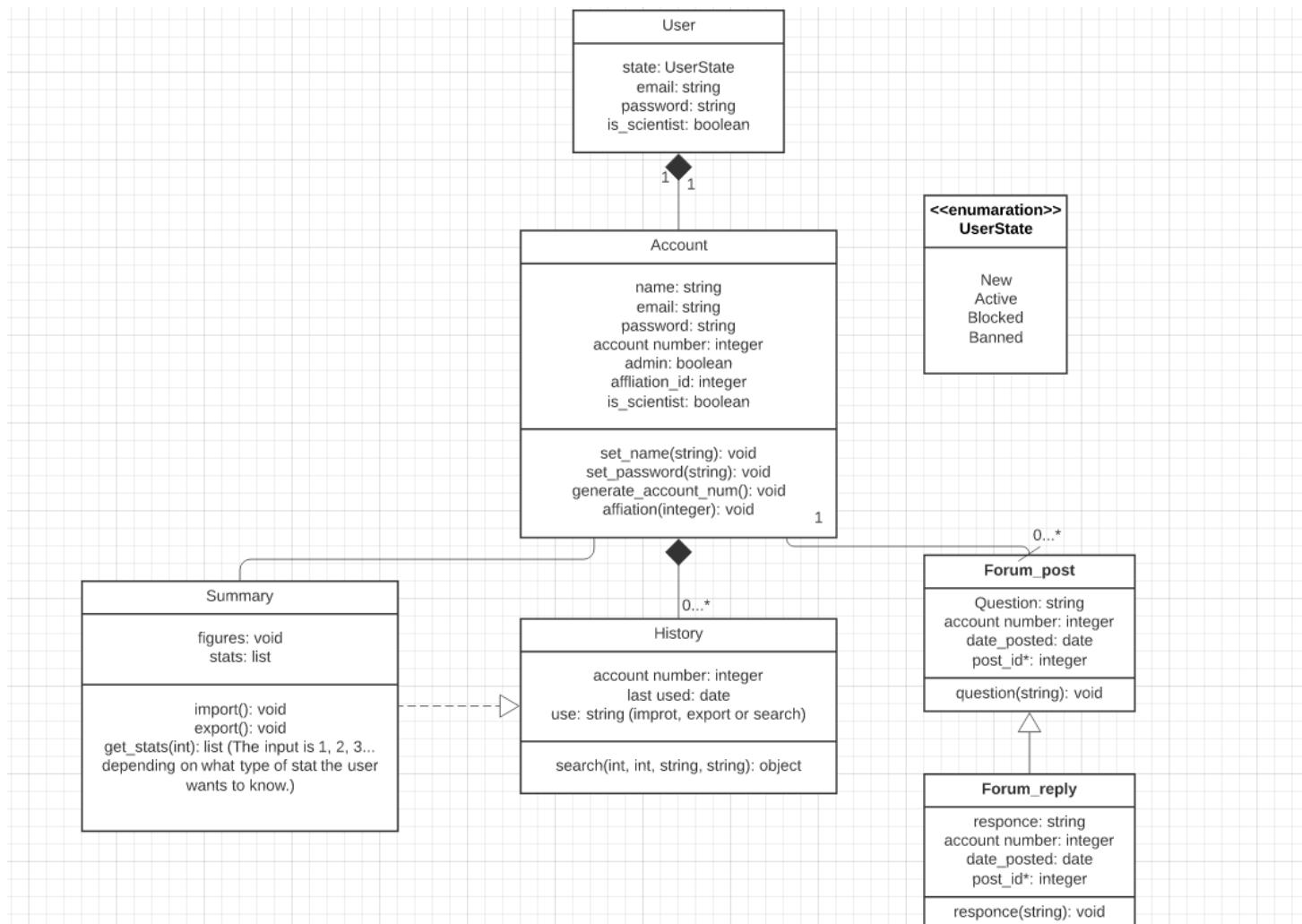
Home forum page:

Routes shown in MVC format

Route	Wireframe view	Controller function
'/'	Home page	home() – returns home page
'/user/login'	Login page	login() – returns login page that allows user to login
'/user/account'	Sign-up page	signup() – returns sign-up page that allows user to create new account
'/user/dashboard'	Dashboard	dashboard() – returns the user's dashboard
'/user/<username>'	Account management page	user_account_management() – returns account management page that allows user to see their account details and change their password and other relevant settings (visual, etc)
'/help'	Help page	help() – returns help page
'/user/forum'	Home forum page	view_forum() – returns main forum page that allows user to view forum posts and create their own forum post
'/user/forum/<post_id>'	Sub-forum page	view_post() – allows user to see a specific post and reply to it
'/about'	About page	about() – returns about page
'/user/analysis'	User analysis page	analysis() – returns analysis page that allows user to import dataset

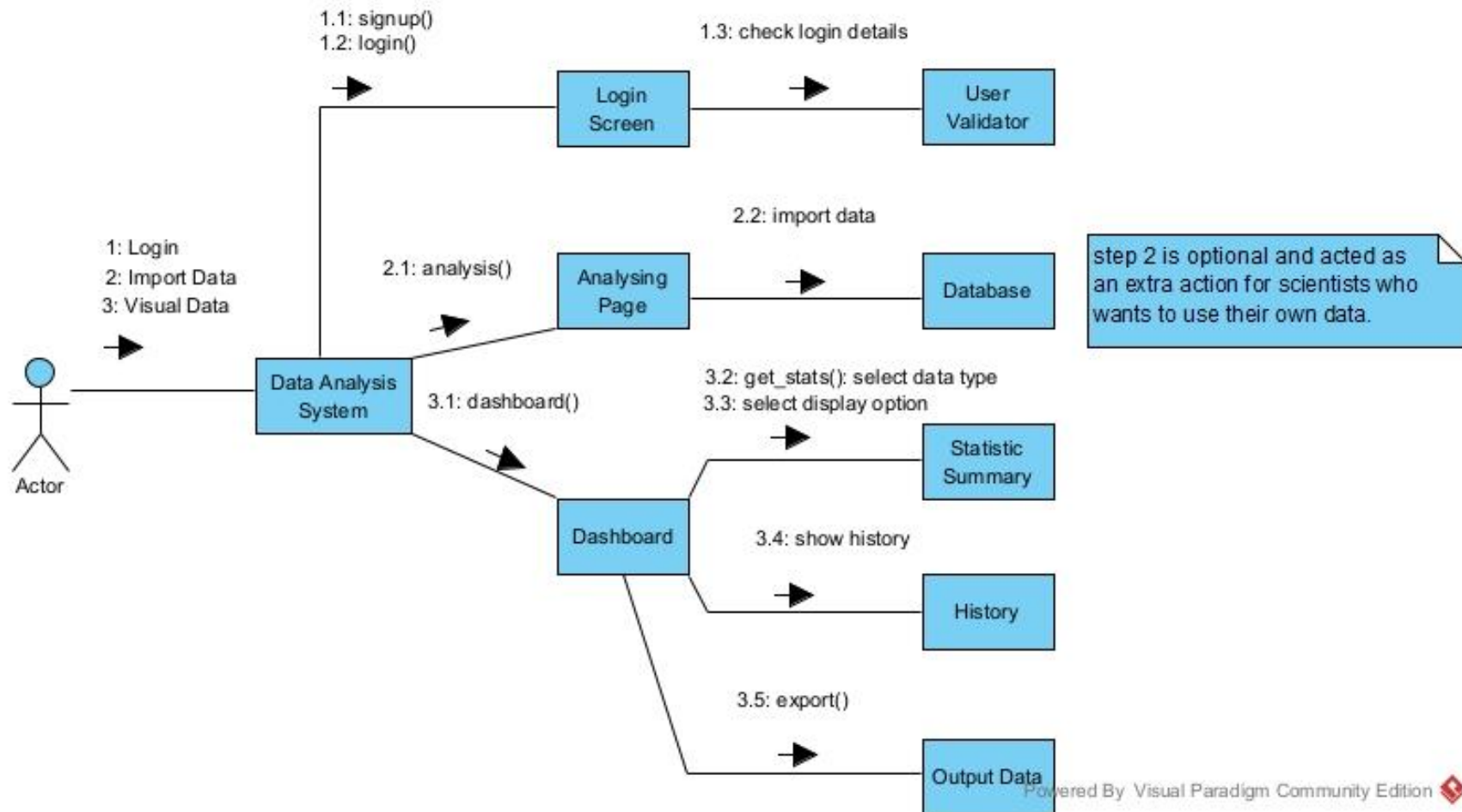
Class diagram showing each class's attributes and methods

A UML class diagram was chosen for the application design section because it clearly shows the structure and functionality of the web application through a common language for designers, developers, and stakeholders. It shows the relationships between different classes and how they will work together, as well as an overview of the functions and attributes within each class. Furthermore, the class diagram will allow the identification of potential design issues early on and will serve as a future reference when developing the application. A UML class diagram was chosen over an MVC because it is more visual and therefore easier to understand for beginner developers.



Communication diagram (structure and flow of the interface)

A communication diagram was chosen to represent the structure and flow of the interface because it is visual (therefore easy for beginner developers) and helps understanding of the wireframes, MVC routes, and class diagram shown earlier in this design document.



Entity relationship diagram (ERD) – relational database design

An ERD gives a visual representation of the data by showing the relationships between the different entities. This better helps understand the requirements of the system, ultimately increasing the efficiency of the database through understanding, identifying, and eliminating unnecessary relationships between entities. Similar to the other UML and communication diagrams, it also helps highlight any early problems with the database structure before writing the codes, as well as allows for better communication between designers and stakeholders regarding the database design.

