**Stage 3 – Prototype**

**1. Statement and justification of choice of prototype**

A horizontal prototype was chosen to outline a broad range of classes and functions that will be used in the aplication as the finer details are still yet to be determined as the project progresses. An evolutionary prototype was chosen as this prototype will be a base upon which the application will built and will evolutionize throughout the project and the classes and functions will expand become more finely detailed. The prototype also includes the frontend (UI) so that the client can get a feel for it and make suggestions so that they are happy with this from an early stage.

**2. Class definitions**

The classes that have been defined in this prototype include:

- **Search:** The search class is in charge of retrieving all the relevant information from the Stations and Route classes in order to find appropriate routes from the departure station to the destination station and return this information to the user.

**- Stations:** The station class contains the names of all the stations and checks if the entered departure and destination stations exist.

**- Routes:** The Routes class retrieves the departure and destination stations and uses the map of connecting routes to find all possible routes. It also retrieves the time and sends the time and routes to the TimeSort class which returns an ordered list of routes available.

**- TimeSort:** The TimeSort class uses the routes and their times to determine the fastest routes. It returns an ordered list of routes that is best suited for the start or end time that the user requests and also the by the duration of each route.

**- Person:** The Person class contains a list of attribute that each person will have, such as their username, name, surname, email and password.

**- User:** The User class inherits the attributes of the Person class and also contains an attribute called history which contains all their previous searches.

**- Admin:** The Admin class also inherits the attributes of the Person class and has another attribute called access which is set to True which allows them to make changes to the database, web page and backend code.

**3. Class member functions**

**- Search:** The Search class has a findRoutes function which takes all the information that the user inputted to check that the stations exist by calling the findStation function and then use the stations to find the routes and retrieve an ordered list of routes available by calling the getRoutes function and returns this list to the user.

**- Stations:** The Stations class has a findStation function which retrieves a station name as a parameter and checks the array of stations to see if the requested stations are in the database. It returns a True or False value depending on if the station was found.

**- Routes:** The Route class has a getRoutes function which will use the departure and destination stations to find available routes in the route map. It will then send either the start or end time as well as the array of available routes to the orderShortestStart or orderShortestEnd function which will return an ordered list of routes.

**- TimeSort:** The TimeSort class has a orderShortestStart function and orderShortestEnd function which takes the array of routes and finds the times for the routes supplied. It will then find the times that suit either the start or end time that was supplied and order the routes from times that are closest to the time supplied and with the shortest duration. It will then return this ordered list of routes.

**- Person:** The Person class has getName, getUsername, getSurname, getEmail and getPassword functions that return the name, username, surname, email and password respectively.

**- User:** The User class has a getHistory function that returns the history of user for increased functionality and ease of access for the user. It also has the inherited functions of the Person class as listed above.

**- Admin:** The Admin class has a getAccess function which returns True to indicate that this user is an admin and has access to make changes to the web page, database and backend code.

**4. Class inheritance**

The class inheritance found in this application is that the User and Admin classes inherit the parent class, Person, to retrieve the mutual attributes that the User and Admin classes share. It will also inherit the Person classes functions and the User and Admin classes have extra attributes that are unique to their respective classses as well as their own functions. Namely, the User and Admin classes will both have the attributes found in the Person class which include: name, surname, username, email and password. The User class, however, has a history attribute which returns its previous searches and the Admin class has an access attribute which gives it additional functions to make changes to the application to which the User class does not have access.

**5. Scope**

A horizontal prototype was created and have been able to create a basic web page that allows the user to sign-up or login, whether they have created an account or not, and allows the user to input the required information for the application to make the relevant search. A few basic tests have been run on the login and sign-up pages. The database has also been created containing most of the information from the train schedule PDFs with the Area North PDF still needing to be transferred and the multiple routes from one station to another to be identified. The backend of the application including the main classes and functions within in these classes has been designed to show the responsibilities of these classes and how they will interact with one another to retrieve the information that the user is requesting.