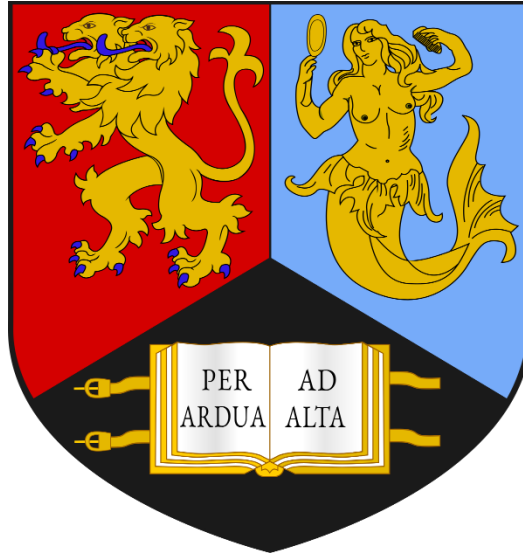


Software Engineering

Group 23



Design of an Interactive Map of Birmingham City for the 2022 Commonwealth Games

Thomas Armstrong-2179997-TLA097

Andrei-Robert Cuzenco-2170987-AXC238

Amelia Fox-2140770-AXF070

Maria Merkulova-2186757-MXM1303

Daniel Perry-2166658-DXP058

Iris Aksunger-2228778-IDA078

Radoslaw Grzemiszka-2128246-RXG046

Daisy Kilbey-2174771-DXK074

Alexander Nutt-2165043-AXN043

Alexander Parker-2184460-ADP060

Afarin Izadi Dehkordi-2253985-ASI085

A. Requirements Engineering

A1. Introduction

BruMap application provides an interactive map during the Commonwealth Games 2022. The aim of BruMap is to explore Birmingham by foot while also completing milestones and obtaining rewards. BruMap will promote a healthier lifestyle and unlock facts about both Birmingham and the Commonwealth's history.

Scope of the System

The scope of the map is limited to the Birmingham area, with the possibility of extending it to nearby locations included in the Commonwealth Games 2022. BruMap will colour code places by category and include locations of the games, nearby restaurants, sights, and landmarks. Mobile accessibility guidelines will be followed during the design and development stages. BruMap will be available to anyone; however, users must register to benefit from the rewards or functionalities such as the milestone tracker. Otherwise, the users will not be able to fully interact with the map.

In the home screen, the following options will be presented:

- Login
- Register
- Continue as Guest
- Reset password

When the user is registering, they will need to provide an email address, create a username and a password. A pop-up window will then ask for access to track the user's location. If the user allows location access, then the milestones and rewards functionality will be available. The user's location is then tracked, and important locations nearby are displayed. If the user does not allow location services, or continues as a guest, milestones and rewards functionality will not be available and the locations displayed may be less relevant to user's location.

Milestones and rewards will be available during the Commonwealth Games in Birmingham. There will be two types of milestones the users can complete:

- Based on steps taken by the user
- Number of landmarks visited

The first milestone will promote a healthier lifestyle by encouraging users to be more active during the games. For every 1000 steps walked, a new trivia fact about the Commonwealth's history will be unlocked. A scoreboard feature will be available to display the most active users on the app. As the user's username will be displayed to all users, all data collected must be stored according to GDPR.

Another milestone will be visiting landmarks in Birmingham. The users need to take a selfie at each location they visit and share them on their social media accounts with their unique tracker ID. The users with at least 4 selfies at 4 landmarks will enter the prize pool raffle. A grand prize will be awarded to one of the users that visited all 20 locations. The prizes might include 1 year full-inclusive gym membership, fitness equipment,

Commonwealth Games merchandise, etc. The grand prize will be a super ticket for the next games. It is assumed the app will have sufficient sponsorship deals.

BruMap will present 3D pinpoint locations with specific colours and symbols (to account for colour blindness) representing subcategories:

- Yellow & Birmingham 2022 logo = Commonwealth game venues
 - Game venues will be displayed
 - Once a particular icon has been selected, relevant information will be presented such as:
 - details of the games shown at that venue
 - System will push any changes to events to users
- Red & knife/fork symbol = Restaurants
 - Once a particular icon has been selected, a link to their website will be displayed
 - Certain restaurants may be takeaway only, therefore, multiple food delivery apps will be linked for convenience purposes (partnerships with such apps will provide funding for app development)
- Green & transport symbol depending on the type of transport = Transport
 - Once a particular icon has been selected, it will display the timetable of the transport (link to website)
 - System will push any delays/updates to all users
- Blue & star symbol = Landmarks
 - Once a particular icon has been selected, the name of the landmark and an image will be displayed
 - Link to websites of landmarks
- Pink & Shopping bag symbol = Shop

A filtering system will allow users to select which pinpoints they wish to see; this will avoid overcrowding of pins on the map. External data on popularity of each location will be used to set a priority for each pin; when zoomed out, only the highest priority pins will be shown, becoming more niche as the user zooms in.

The application must be scalable to allow for many users accessing the app at the same time without the app crashing.

It is assumed there is sufficient back-up provision for application continuity in the case of main server or database errors.

The application will not cover payments, food delivery or the actual sending of password reset links however it will send requests to the relevant sub systems to perform these actions.

The application will record the steps each user has walked. However, it is assumed there will be a relevant subsystem that can perform this functionality.

A2. Requirements Analysis

Functional Requirements

1. Registered User

- 1.1. A new user should be able to register with an email and username and password
- 1.2. A user that is already registered can log in using their email and password
- 1.3. The password must contain both uppercase and lowercase characters
- 1.4. Each user's username does not need to be unique
- 1.5. A user should not be able to create a new account with an email that is already registered
- 1.6. The user should be able to reset their password by providing their email
 - 1.6.1 If the provided email is registered, an email should be sent with a link to reset the password
 - 1.6.2 If the provided email is not registered, the user should be taken to the register screen. No email will be sent.
- 1.7. Each user's account should hold the user's information
 - 1.7.1. The account should track the user's milestones
 - 1.7.2. The account should hold the user's rewards
 - 1.7.3. The account should track the total distance the user has walked

2. Guest User

- 2.1. Guest users must be able to view the map
- 2.2. Guest users must not be able to complete milestones
- 2.3. Guest users must not be able to collect rewards

3. Interactive Map

- 3.1. The map must track and show the user's location if they allow it
 - 3.1.1. The map must not track the user's location if location services are turned off
- 3.2. The map must show points of interest
 - 3.2.1. The map should show public transportation
 - 3.2.2. The map should show local restaurants
 - 3.2.3. The map should show local shops
 - 3.2.4. The map should show the venues for games events
 - 3.2.5. The map should show sights and landmarks
- 3.3. The points of interest should be colour coded and have a specific symbol
 - 3.3.1. Public transportation should be shown in green with transport symbol that varies depending on type of transport
 - 3.3.2. Restaurants should be shown in red with knife/fork symbol
 - 3.3.3. Shops should be shown in pink with shopping bag symbol
 - 3.3.4. Venues should be shown in yellow with Birmingham 2022 logo
 - 3.3.5. Sights and landmarks should be shown in blue with star symbol
- 3.4. The user should be able to click on points of interest to view more information
 - 3.4.1. A link to each restaurant's website could be displayed
 - 3.4.2. The events taking place at venues should be displayed
 - 3.4.3. The timetable for public transport could be displayed
 - 3.4.4. A link to each shop's website could be displayed

- 3.4.5. Information about each landmark should be displayed
- 3.4.6. The image of each landmark should be displayed
- 3.4.7. If a landmark has a website the website is displayed
- 3.4.8. The name of each point of interest will be shown
- 3.5. the user should be able to filter the points of interest shown on the map
 - 3.5.1. The user should be able to choose to view only restaurants
 - 3.5.2. The user should be able to choose to view only venues
 - 3.5.3. The user should be able to choose to view only shops
 - 3.5.4. The user should be able to choose to view only sights and landmarks
 - 3.5.5. The user should be able to choose to view only public transport
 - 3.5.6. The user should be able to apply multiple filters at once
- 3.6. There should be 20 designated main landmarks
 - 3.6.1 when a user has visited all 20 main landmarks, they can enter to win the grand prize
- 3.7. The user should be able to zoom in and out of the map
 - 3.7.1 the map should be able to zoom into 10 meters
 - 3.7.2 the map should be able to zoom out to 5 kilometres
 - 3.7.3 a maximum of 15 pins will be shown on screen at a time
 - 3.7.4 when there are more than 15 pins in the radius of the map, the ones shown will be decided based on their priority
- 3.8 Once a user has visited a landmark or sight, it should be marked as visited on their map
 - 3.8.1 The pin for the visited landmark or sight should change from a star to a checkmark to indicate it has been visited

4. Milestones

- 4.1. There must be milestones for distance travelled by the user
 - 4.1.1. Each 1000 steps that a user completes should unlock a fact about either the Commonwealth games or Birmingham
 - 4.1.2. When a user first unlocks a fact, they will be notified via a pop-up on their screen.
 - 4.1.3. Users can view all their unlocked facts on their profile page.
 - 4.1.4. There should be a scoreboard showing the users with the most steps
- 4.2. There should be milestones for locations visited
 - 4.2.1. A user has visited a location when their GPS location is within 10 meters of the landmark
 - 4.2.2. after visiting 4 locations, the user is entered into the prize draw
 - 4.2.3. after visiting 20 locations, the user is entered into the draw for the grand prize.
The 20 locations are:
 - 4.2.3.1. Birmingham Back to Backs
 - 4.2.3.2. Winterbourne House and Garden
 - 4.2.3.3. Gas Street Basin
 - 4.2.3.4. Museum of the Jewellery Quarter

- 4.2.3.5. Aston Hall
- 4.2.3.6. Brindleyplace
- 4.2.3.7. Soho House
- 4.2.3.8. St. Philip's Cathedral
- 4.2.3.9. St. Martin
- 4.2.3.10. Bullring. & Grand Central
- 4.2.3.11. Birmingham Museum and Art Gallery
- 4.2.3.12. Cadbury World
- 4.2.3.13. Black Country Living Museum
- 4.2.3.14. The Birmingham Botanical Gardens
- 4.2.3.15. The Barber Institute of Fine Arts
- 4.2.3.16. Old Joe
- 4.2.3.17. Lapworth Museum of Geology
- 4.2.3.18. Hall of Memory
- 4.2.3.19. Victoria Square
- 4.2.3.20. Thinktank. Birmingham Science Museum
- 4.3. The user should be able to view their completed milestones
- 4.4. If location services are turned off, the user cannot unlock any milestones

Non-Functional Requirements

1. Performance

1.1. Workload and Concurrency

- 1.1.1. The system will be able to process 20 000 users at any given time
- 1.1.2. The system will be able to process 40 000 requests per second
- 1.1.3. The system will be able to render 15 pins at once
- 1.1.4. Data consistency is maintained at all times.

1.2. Response Time

- 1.2.1. The client will be able to respond to the input within 1 seconds
- 1.2.2. The server will respond within a reasonable amount of time
- 1.2.3. The location will be updated every 2 seconds

1.3. Storage Constraints

- 1.3.1. The system will be able to store the data of 1 million users on the server
- 1.3.2. The system will be able to store the data of all the locations on the server
- 1.3.3. The system will be able to store the top 100 users on the leader board

2. Usability

2.1. Map Readability

- 2.1.1. The points of interest have different shapes and symbols, for colourblind people
- 2.1.2. The system will have a clear distinction between the map's background (road's, grassland areas, housing) and the points of interest, which will be clearly outlined with a unique shape and colour
- 2.1.3. The map will be rendered enough, so that the user will be able to see around them without a problem

2.2. Text Readability

- 2.2.1. The client will have text font of at least 14px, in order to make everything readable
- 2.2.2. The client will allow for a text-to-speech option to allow the use of the app for users who are unable to read, which is to be provided externally
- 2.2.3. The client will have an option for a language choice, which is to be provided externally
- 2.2.4. The client will have black text on white background

2.3. General ease of access

- 2.3.1. The client will have appropriately sized input sections (buttons, labels, input boxes)
- 2.3.2. The client will not be too bright/dark, having a pleasant colour scheme
- 2.3.3. The client will be easy to operate, and easy to learn

3. Security

3.1. User Privacy

- 3.1.1. The system will ensure that user location is not shared with other users or stored on the server and will not be accessible to anyone other than the user themselves, asking the user prior to using the map whether or not they agree to share it
- 3.1.2. The system will ensure that user's email is not shown on their profile
- 3.1.3. The system will display usernames on the leader board rather than first and last names

3.2. Data security

- 3.2.1. The system will ensure that all data on the server is encrypted using a minimum of AES 128bit
- 3.2.2. The system will ensure that all communications between the server and the client are encrypted using RSA 2048 bits

3.3. System security

- 3.3.1. The server will have the appropriate security measures in place (DDOS protection, firewall, only necessary data will be shared for server responses)
- 3.3.2. The client will have the necessary precautions in place to prevent tampering and manipulation of the app/system (The system will prevent users from reverse engineering the app)

4. Reliability

4.1. Data Reliability

- 4.1.1. Database must be normalized to Boyce-Codd normal form
- 4.1.2. The client must accurately track the user's distance travelled to the nearest metre
- 4.1.3. The client must accurately track the user's location within 2 metres (considering good service), and within 10 metres otherwise
- 4.1.4. The leaderboard data must be updated every minute
- 4.1.5. The data stored on the server will be up to date (both user data and data on different locations)

4.2. System Reliability

- 4.2.1. The system must be available 99% of the time 24/7
- 4.2.2. The system must sync databases and back-up databases every 24 hours
- 4.2.3. The system must have a low crashing probability (1 in 1000 hours)

4.2.4. The system will perform a backup every 24 hours

5. Maintainability

5.1. The system will ensure that the database will maintain a Boyce-Codd Normal form

5.2. The system will store data in such a way that will allow easy scalability

5.2.1. The system should be designed such that additional databases and servers can be managed easily.

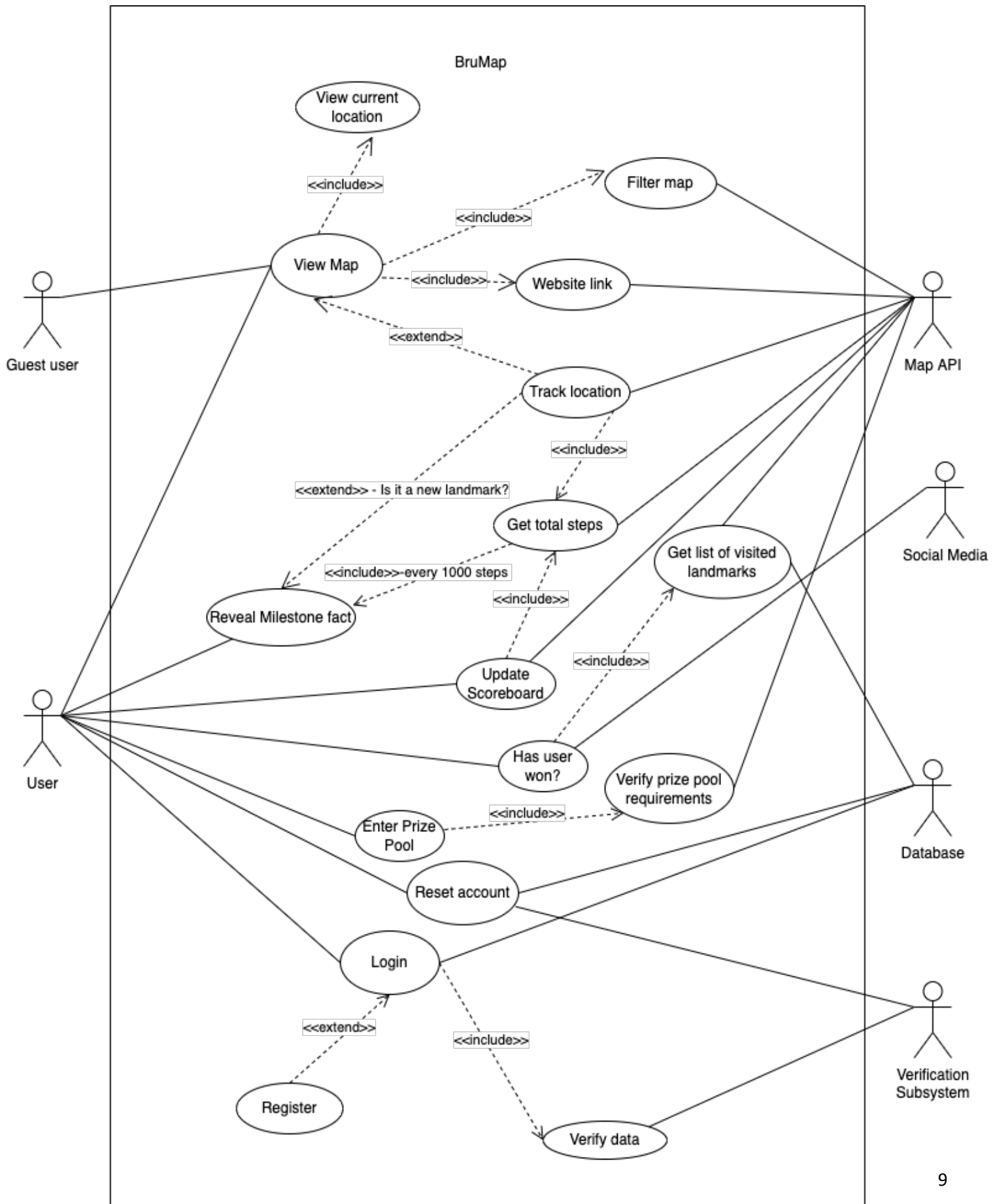
5.3. The system will not change the specific protocols or APIs that the client and server use to communicate

5.4. The system will be able to easily update data on the database

5.5. The system will suspend any distance tracking, achievements and promotion of the Commonwealth Games a week after the game have finished

B. Software Design with UML

B1. Use Case Diagram



B2. Documented Use Cases

First Use Case

Use Cases Descriptions

Filter to see the restaurants only

Precondition

1. The user has installed the app onto the phone.
2. The device has an active internet connection or data.

Flow of events

1. The user opens the app.
2. The user is prompted with the option to either log in, register, reset password or view as guest user.
 - 2a. The user chooses to view as a guest user.
 - 2b. The user chooses to register a new account, so they are prompted to fill in a form including username, password and email. The submitted form is sent to the verification subsystem to be checked for invalid data.
 - 2ba. If the email is already registered provide an error to the user saying “The email is already registered. Please choose another email or log in.”
 - 2bb. If the email isn’t registered and the provided data is valid, add a new user into the database and provide a success message to the user saying: “Account is created.” and prompt the user to log in.
 - 2bc. If the provided data is invalid, provide an error message saying: “The provided data is invalid please try again.”
 - 2c. The user chooses to log into their account by entering their email and password. The submission is sent to the database and checked for correct email-password match.
 - 2ca. If the email is not in the database, provide an error message saying: “User is not registered.”.
 - 2cb. If the email exists but the password is incorrect provide an error message saying: “The password is incorrect. Please try again or reset your password.”.
 - 2cc. If both email and password are correct log the user in and enable all the functionality of a registered user.
 - 2d. The user chooses to reset password. The user is directed to the password reset sub-system that manages passwords.
3. The map and options are displayed to the user.

- 3a. If the user has enabled GPS services, the locations most relevant to the user at their position are displayed on the map.
- 3b. If GPS services are disabled, the map displays a general map of Birmingham centred on Birmingham city centre.
4. The user filters the points of interest by only viewing restaurants, the map view is updated to display the restaurants only.
 - 4a. If the filtration is unsuccessful, the user is prompted with the relevant error message.
 - 4b. If the filtration is successful, only the restaurants are displayed on the map.
5. The user clicks on one of the restaurants displayed on the map, the name of the restaurant, its address, and a link “View Details” is displayed on the user’s screen.
 - 5a. If the user clicks on the “View Details” option. The name of the restaurant as well as the link to that restaurant’s website is displayed. The user can open this link in their local browser if desired by clicking the link.
 - 5b. If the user’s choice of restaurant does not have a website, the “View Details” functionality displays the name of the restaurant.

Postconditions

1. The app has successfully filtered the points of interest on the map.
2. The user was able to click and interact with their choice on the map.

Second Use Case

Use Cases Descriptions

The user can visit a designated landmark.

Precondition

1. The user has a registered account on the app.
2. The user has logged into the app with a registered account.
3. The verification subsystem has confirmed the user’s login details.
4. The device has an active internet or data connection and has enabled GPS data.

Flow of events

1. The user opens the app.
2. The user opens the map view and filters for designated landmarks.
3. The user then chooses a landmark and is provided with the full name, the address and (if it exists) the website of the registered landmark.

4. On arrival, the map updates the UI and marks the designated landmark as “visited”.
 - 4a. If the user has visited four locations including the current landmark, then prompt the user with a message that asks them if want to enter the prize pool.
 - 4aa. The user chooses not to enter the prize pool.
 - 4ab. The user accepts. The account information will be added to the prize pool raffle.
 - 4b. If the user has not reached four locations yet, then the map will just update the location as visited.

Postconditions

1. The app has successfully marked the location as visited in the app.
2. The app has checked the requirements for entry to prize pool raffle and added the user if eligible.

Actors: User, Map API, Social Media, Database

B3. Scenarios

Scenario 1 – First Use Case

User Antonio would like to find a restaurant nearby. He opens the app and is shown the options to log in, register, reset password or view as guest user. Antonio is unsure if he has made an account therefore he decides to select the register option. Antonio is prompted to fill in a form that takes their username, email address and password. Antonio attempts to register but is met with the message: “Email is already registered. Please choose another email or log in.” Antonio then selects the reset password option as he cannot remember his password. An email is sent to Antonio with a link to reset his password which he follows. Antonio successfully resets his password. Now he is shown the map view and options. Antonio has GPS services switched on his device and therefore he is shown the nearest locations to his position on the map. Antonio would like to find a restaurant nearby so he uses the filter functionality to filter the map specifically for restaurants which are then shown displayed to him on the map. Using the map, Antonio then walks to the nearest restaurant “Wayland’s Yard”. While walking, Antonio decides to check the opening times for the restaurant, so he selects the “View Details” function, but he is only met with the name of the restaurant. Wayland’s yard has no website.

Scenario 2- First Use Case

User Sam is looking for any local restaurants near him. He opens the app and is shown options to log in, register, reset password or view as guest user. Sam logs into the app using his email and password. Sam attempts to log in but is told “The password is incorrect. Please try again or reset your password.” Sam has typed his password incorrectly, so tries to log in again which is then successful. Sam is then shown the map with every point of interest and landmark shown locally to him displayed. Sam chooses to filter the map to just restaurants. Now Sam is shown the restaurants close to his position on the map. Sam decides he would like to eat at the “Grand Central Kitchen” and would like to check the opening times. He selects the “View details” option on the restaurant’s icon and is shown a link to the restaurant's website where he finds the opening and closing times.

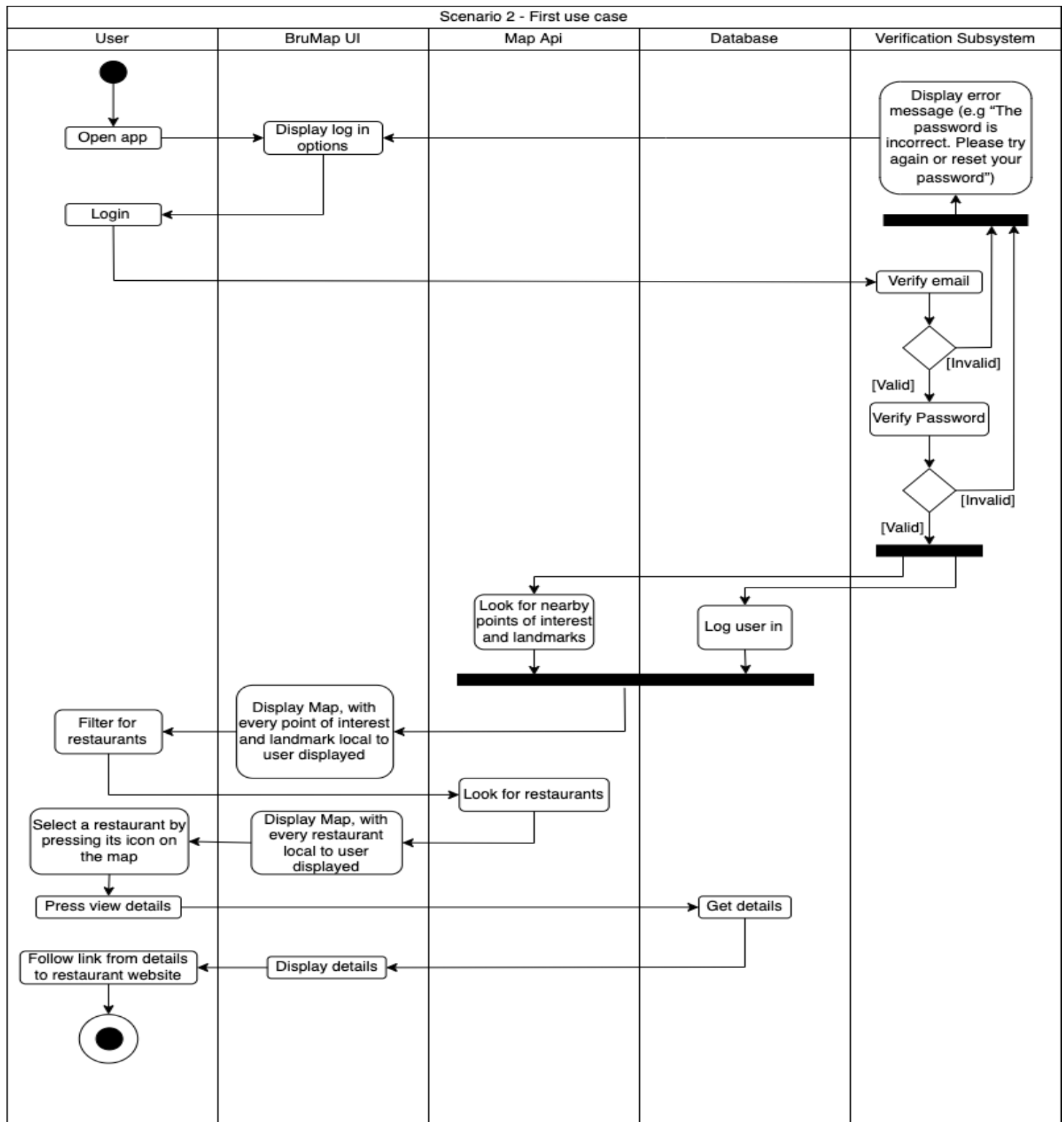
Scenario 1 – Second Use case

User Ava decides to visit the Birmingham Botanical Gardens. She opens the app and is immediately prompted with the map view. She filters for landmarks and clicks on the botanical gardens. Ava decides to look on their website for the opening times before going and is prompted by a fact about the garden. After seeing the landmark is open, she takes the bus. The location is now updated as visited and Ava only needs three more locations to enter the prize raffle.

Scenario 2 – Second Use case

User O'Malley is thinking about visiting the St. Phillip's Cathedral. He opens the map and uses the filter to view only landmarks and chooses the cathedral. Unfortunately, after looking at their website, they are closed for today so instead he decides to go to the Bullring Station. After looking at ways of getting there through an external application, he decides to cycle to the destination. After arriving he is prompted with a fact about the location and because of reaching four locations he can enter the prize raffle.

B4. Activity Diagram



B5. Class Analysis

Noun/Verb Analysis

We have applied noun/verb analysis from our requirements to determine possible classes and objectives for our application. Each noun in the requirements document was considered as a potential class or attribute.

Candidate Classes (Nouns)

Word/Phrase	Accepted	Reason
User	User	This can be used for a class to store all data related to each specific user
Account	Account	This can be used as a class to store purely login data for each user
Screen	Screen	This class displays information to the user, both for logging in and for usage of the app
Username	userName	Attribute of User Class
Email	accountEmail	Used as an attribute for the Account class
Password	accountPassword	Used as an attribute for the Account class
Milestones	Milestone	This can be used as a class to store all data related to both distance and landmark milestones
Information	No	Too general
App	No	Too general, refers to system as a whole
Rewards	milestoneRewards	Used as an abstract subclass for the Milestone class,
Steps	userSteps	Attribute of User
Pedometer	No	Outside scope- within Map API
Map	MapView	Used as a class to store all data shown on the screen to the user
Birmingham	No	Irrelevant to scope
Location	userLocation	Attribute of User that feeds into MapView
Location services (GPS)	No	Functionality used however not in scope

Points of Interest	pointOfInterest	Used as a class that links to MapView that stores all points to be shown on the map
Restaurant	restaurant	Sub class of pointOfInterest. Stores information specific to a restaurant. i.e. - menu
Public transportation	publicTransport	Sub class of pointOfInterest. Stores information specific to any type of public transportation i.e. timetable
Shops	shop	Sub class of pointOfInterest. Stores information specific to shops i.e. link to website
Venues	venue	Sub class of pointOfInterest. Stores information specific to Commonwealth venues i.e. events shown
Sight	No	Too similar to Landmark
Landmark	landmark	Sub class of pointOfInterest. Stores information specific to the app-defined landmarks, i.e. picture
Colour (green, yellow, red, pink, blue)	colourPOI	Attribute of pointOfInterest
Symbol (Knife/fork, shopping bag, Birmingham 2022 logo, star, transportation)	symbolPOI	Attribute of pointOfInterest
Info	No	Too general to be an attribute
Menu	No	Included within linkPOI
Rating	No	Included within linkPOI
Link	linkPOI	Attribute of pointOfInterest, gives link to external website showing information
Events	No	Included within linkPOI
Timetable	No	Included within linkPOI
Website	No	Same as linkPOI

Facts	landmarkInfo	Attribute of landmark, stores information about each landmark which is shown when you click on the landmark on the map
Name	namePOI	Attribute of pointOfInterest
Image	landmarkImage	Attribute of landmark showing image from e.g. google images that is not copyrighted of the specific landmark, and alt image captions
Commonwealth	No	Event rather than about the system itself
Database	No	Not in scope; use of a database system is however implied to be used within the app
Milestone fact	milestoneFact	Attribute of milestoneRewards
Scoreboard	Scoreboard	Class
Ranking	scoreboardRanking	Attribute of scoreboard
Prize	milestonePrize	Abstract subclass of Milestone
Prize draw	milestonePrizeDraw	Attribute of milestonePrize
Grand Prize	milestoneGrandPrize	Attribute of milestonePrize
Verification subsystem	No	Not in scope; subsystem however is implied to exist and be linked with the DBMS used in order to verify all our data
Data Consistency	No	Outside scope
Concurrency	No	Outside scope
Servers	No	Outside scope
Pop-ups	Yes	Attribute of milestoneRewards

Candidate Operations (Verbs)

Login to account	loginAccount	Method of Account class. Logs in to access the user class
Register a new account	registerAccount	Method of Account class that uses the constructor for Account. Allows the user to register a new account and is thus a method
Create a new account	No	Same as registerAccount.
Reset password	resetAccount	Method of the account class
Be taken to register screen	updateScreen	Method of Screen class. Updates screen as the system requires
Hold user information	No	Database not in scope
View the map	No	Part of updateScreen
Track milestones	trackMilestones	Method of Milestone class. Tracks how many milestones have been completed
Track rewards	No	Is covered within the scope of trackMilestones
Track total steps walked	TrackSteps	Method of User class, if user is a Guest or does not enable pedometer functions, this will always return 0
View the map	No	Map is always visible within the app
Update the map	No	Part of updateScreen
Track user's location	No	Outside the scope
Show user's location on the map	showUser	Method called within updateScreen, updates location of the user
Show points of interest	No	part of updateScreen
Be colour coded	No	Part of showPOI and updateScreen
View more information for point of interest	viewMorePOI	Method called when point of interest is pressed by User
Filter the points of interest	filterPOI	Method called when filter button is pressed, can apply multiple filters
Apply multiple filters	No	Is covered within scope of filterPOI

Unlock a fact	unlockFact	Method in milestoneRewards , called when User completes +1000 steps
Show scoreboard	No	Part of updateScreen
Hide scoreboard	No	Part of updateScreen
Update scoreboard	updateScoreboard	Method called at specific intervals
Enter prize draw	enterPrizeDraw	Method in milestoneRewards, called when user has visited 4 locations
Enter grand prize draw	enterGrandPrizeDraw	Method in milestoneRewards, called when user has visited all locations
Check landmark visitation	No	Outside the scope as this requires searching through user's social media
View completed milestones	No	Part of updateScreen
Hide completed milestones	No	Part of updateScreen
Mark the location of landmark as viewed	viewedLandmark	Method in Landmark called when the user has visited one of the 20 landmarks (this is found out through GPS functionality), changes symbol
Managing databases	No	Outside scope

Responsibility-Driven Analysis

As we have used noun/verb analysis to determine potential classes, we have then furthered those ideas by applying a responsibility-driven analysis to generate CRC cards for each candidate class.

User	
Responsibilities	Collaborators
Maintain all data pertaining to the user <ul style="list-style-type: none"> • Hold the username (link to Account) • Hold and update the steps walked by user • Send the distance to Milestone 	Account Milestone MapView Scoreboard Screen

<ul style="list-style-type: none"> • Hold and update the milestones achieved by the user (through Milestone class) • Send the completed milestones to Milestone as they are completed • Allow the user to view completed milestones when a button is pressed (through the Milestone class) • Show and update the location of the user (through location services access provided by the user) • Show and update the prizes achieved by the user (through the Milestone class) • Send userSteps and userName to the Scoreboard Class if the user chooses 	
---	--

Account	
Responsibilities	Collaborators
Maintain all the data pertaining to the login of the user <ul style="list-style-type: none"> • Hold the username (link to User) • Hold the password • Hold the email • Allow the user to login • Allow the user to register if they do not have an account registered 	User Screen

Milestone	
Responsibilities	Collaborators
Maintain all the data pertaining to the milestones <ul style="list-style-type: none"> • Hold all the available milestone facts that can be sent to a user upon completion of a milestone • Hold the accumulative steps (through the User Class) and sends completed milestones to each user as they walk the correct distance • Enter all of the users who have visited 4 different landmarks into the prize draw (through User) 	User Screen Scoreboard

<ul style="list-style-type: none"> • Enter all users who have visited 20 locations into the grand prize draw (through User) • Send a random Commonwealth fact to the user when milestone is completed (through User) • Send the milestone information to Screen when prompted by system 	
--	--

MapView	
Responsibilities	Collaborators
Display all the data on the screen to the user <ul style="list-style-type: none"> • Show and update the location of the user by changing the view of the map (through the location services) • Show the relevant points of interest on the map (through the pointOfInterest class) • Allow the user to click on a point of interest and view more information (through pointOfInterest) • Allow the User Class to show a landmark as visited (through pointOfInterest) • Send all of the information shown through MapView to Screen 	pointOfInterest User Screen

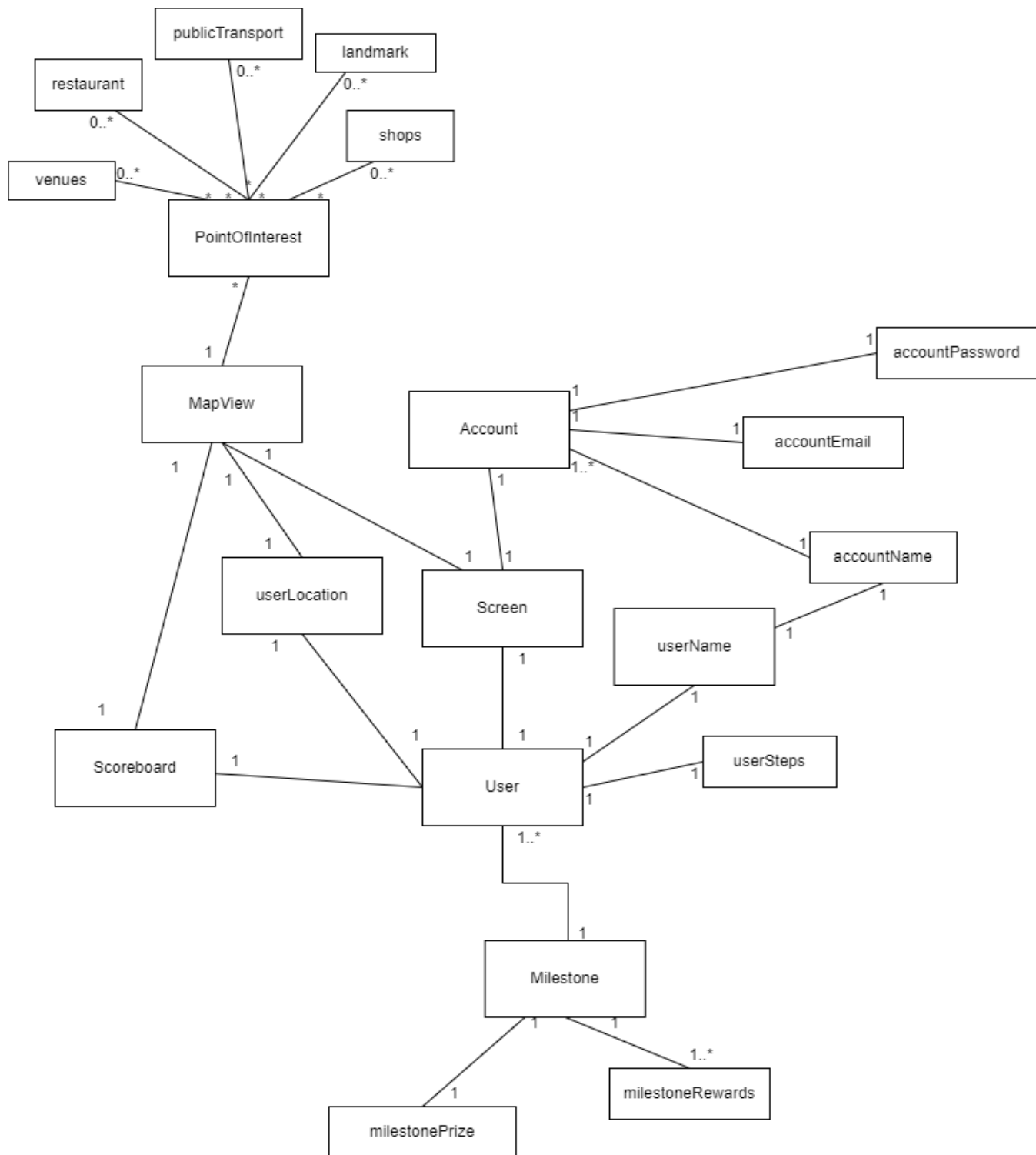
pointOfInterest	
Responsibilities	Collaborators
Display all of the data relevant to the points of interest to the user, and update them when necessary <ul style="list-style-type: none"> • Send all of the points of interest to MapView (if no filtering is used) • Send only points of interest chosen by the user to MapView (if filtering is used) • Send the correct colours and symbols of each point of interest 	MapView

<ul style="list-style-type: none"> • Send the information on a point of interest to MapView when prompted by the system 	
--	--

Screen	
Responsibilities	Collaborators
Display all the data to the user, and update when it necessary. This includes: <ul style="list-style-type: none"> • Register Screen • Map screen • Pop-up scoreboard • Pop-up milestones 	MapView User Scoreboard Account

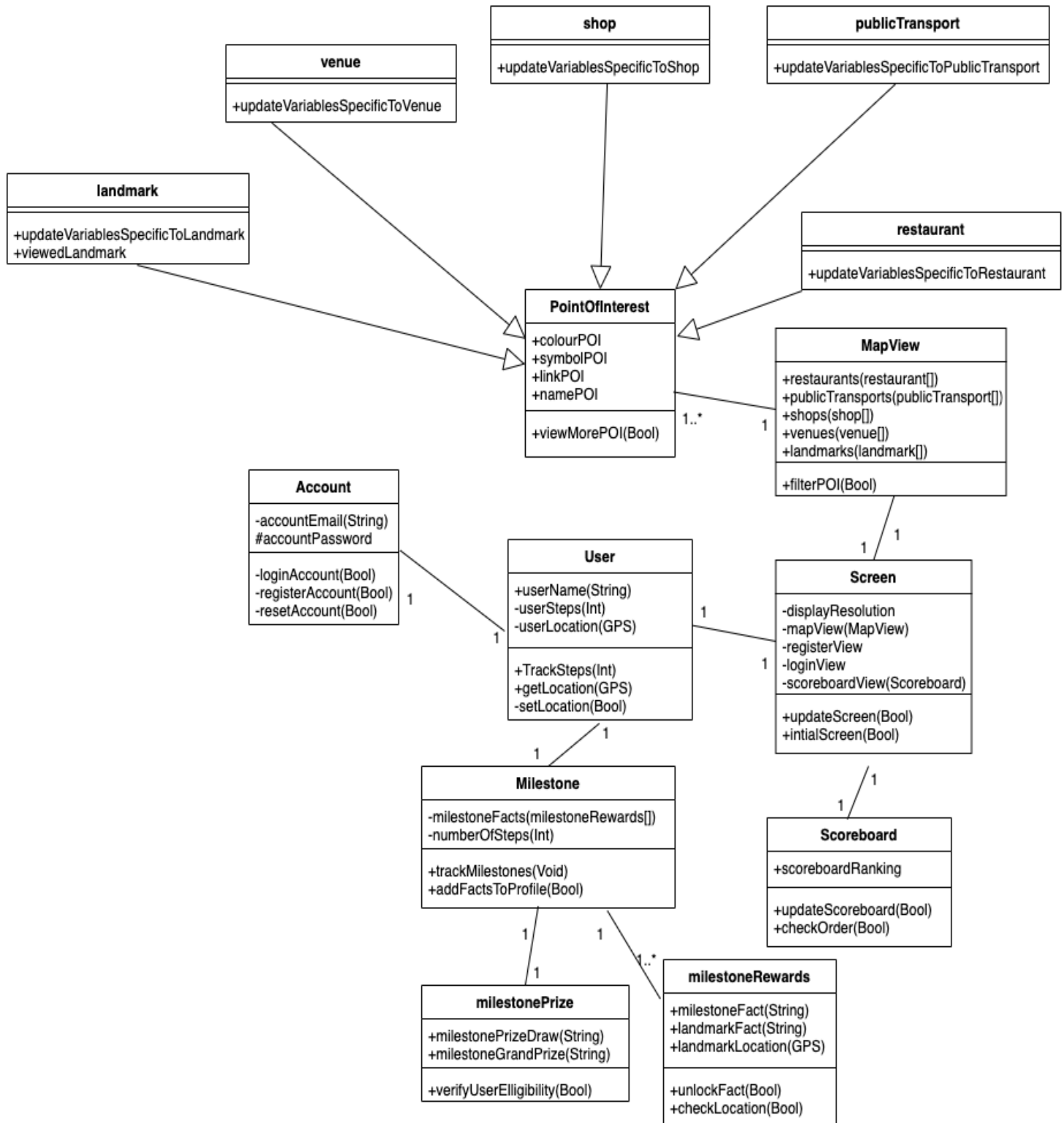
Scoreboard	
Responsibilities	Collaborators
Is responsible for all data pertaining to the step count Leaderboard <ul style="list-style-type: none"> • Update the scoreboard at a regular interval • Collate the scoreboard using the information given through all the users who choose to share data • Send the information to the Screen Class when prompted by the system 	MapView User

First-Cut Class Diagram



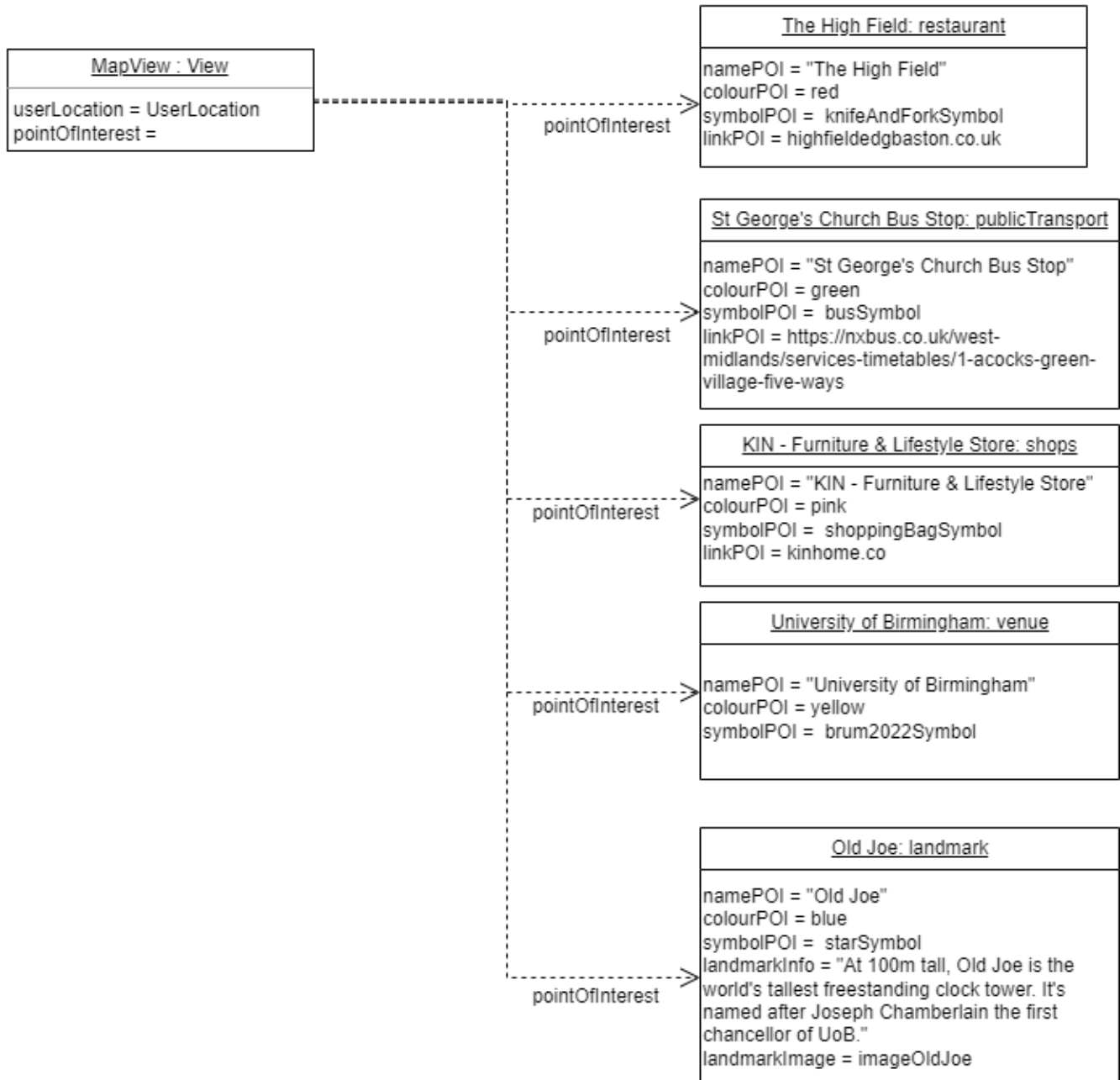
Class Diagram

Class diagram



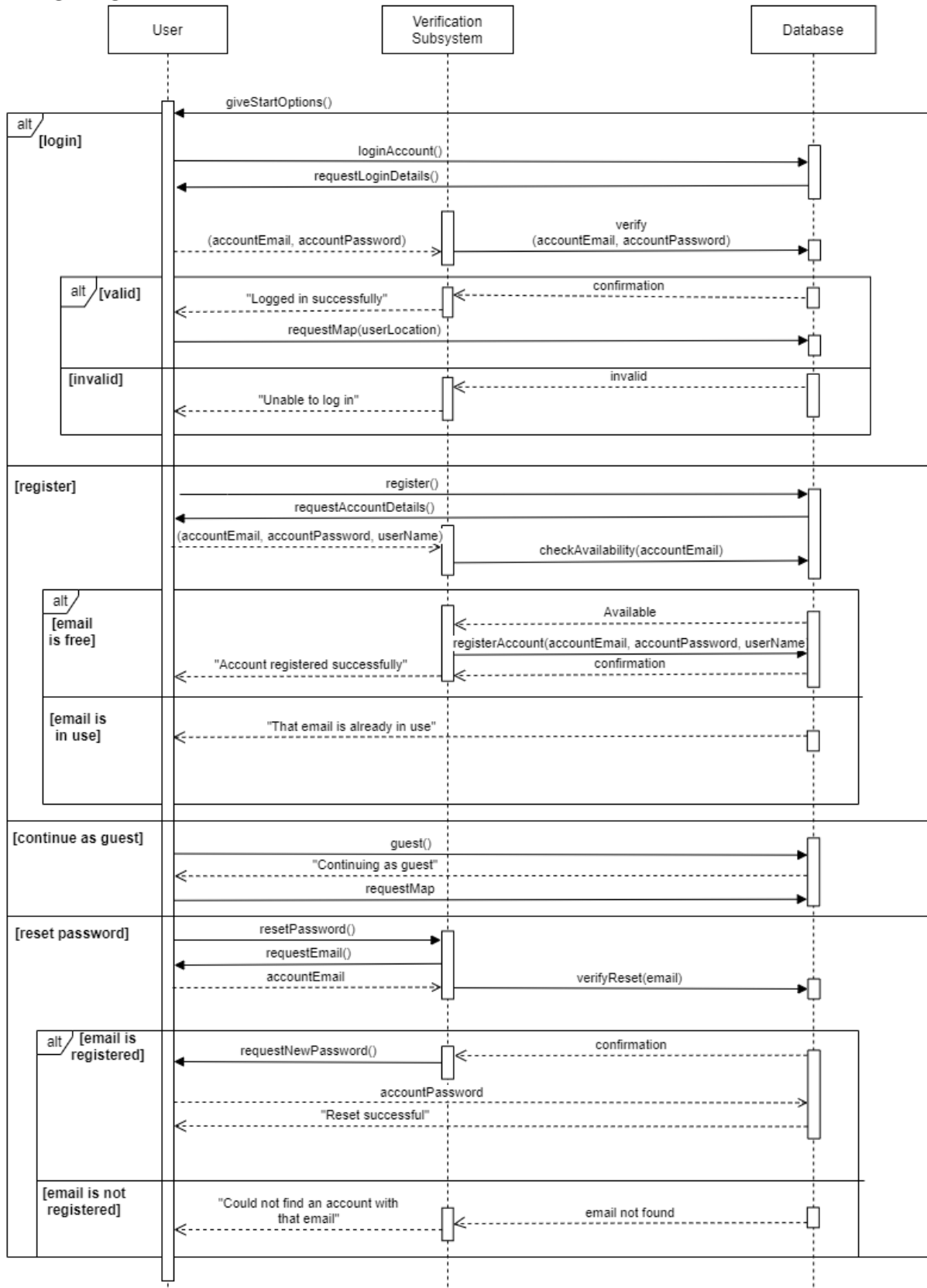
B6. Object Diagram

Object Diagram - Using the Map

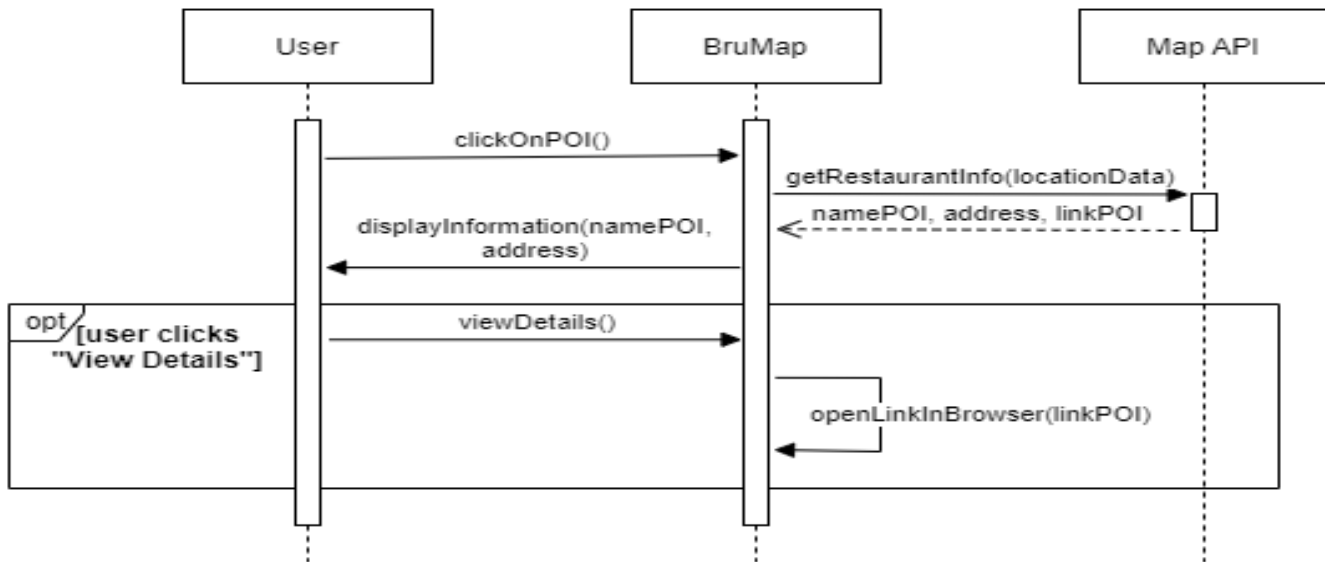


B7. Sequence Diagrams

Sequence Diagram: Login

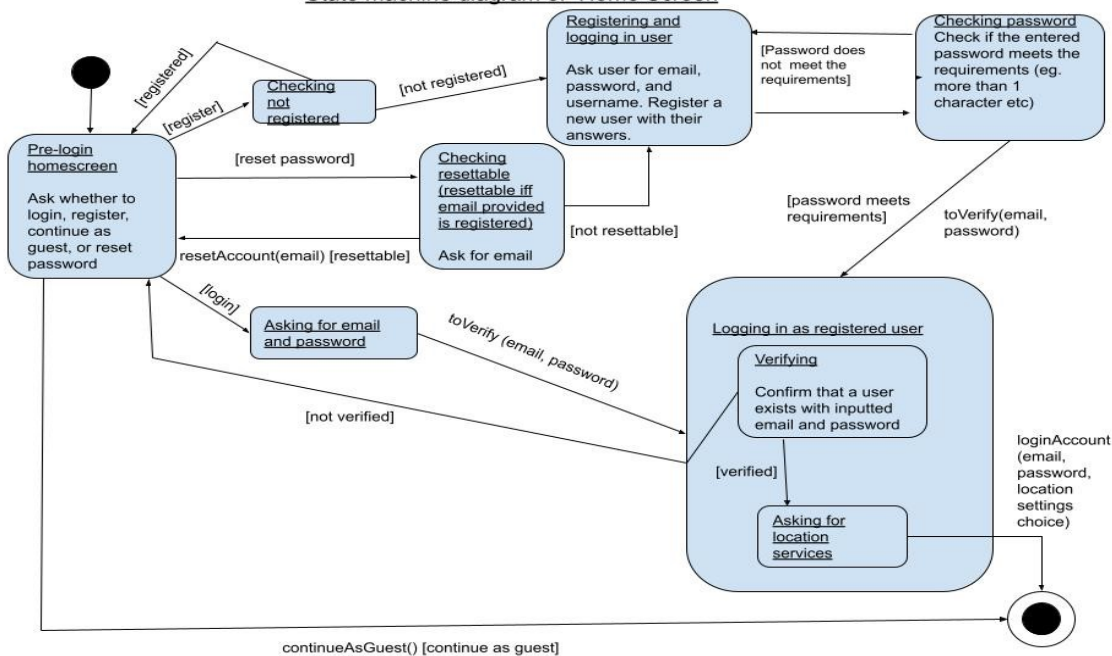


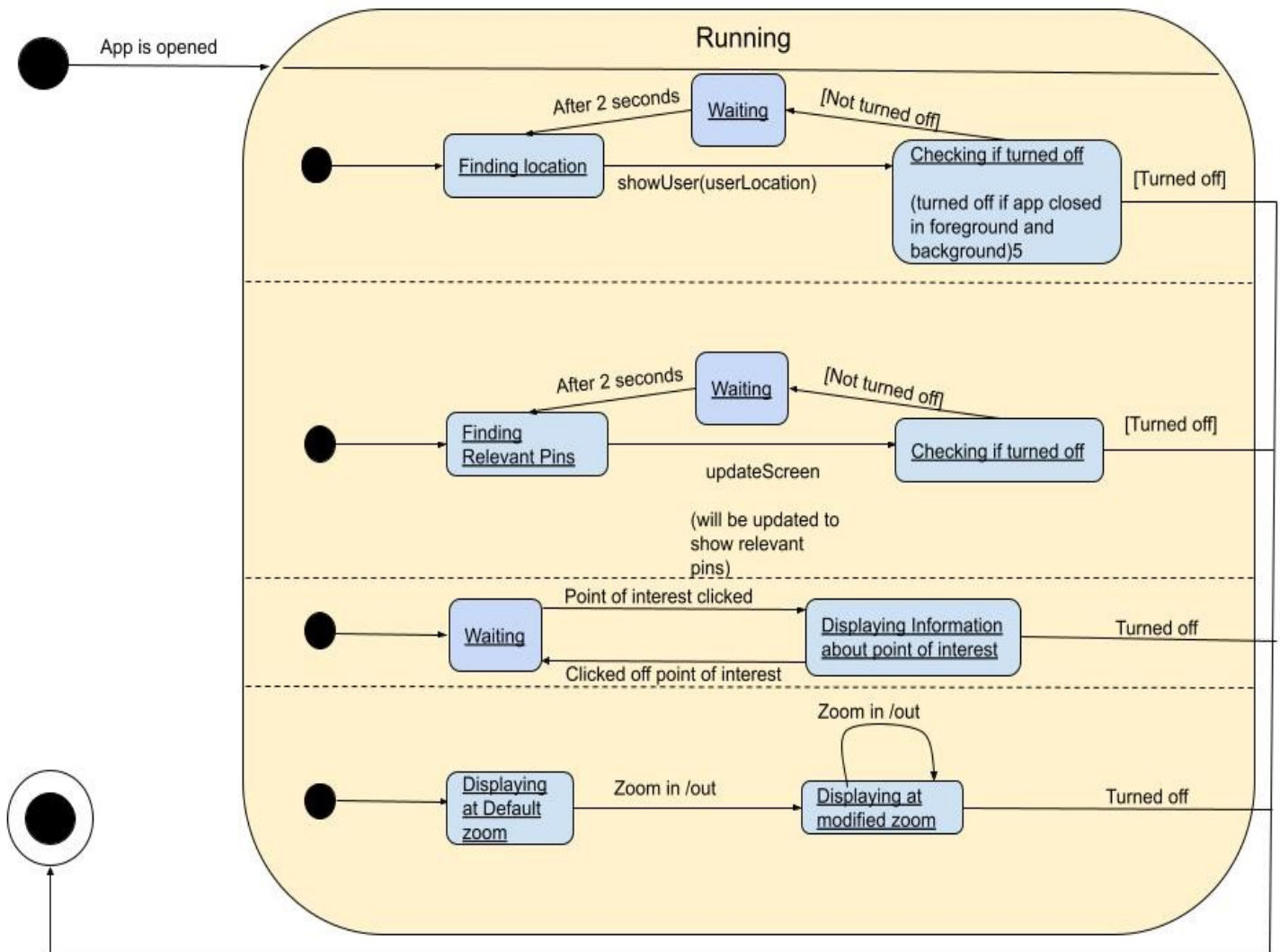
Sequence Diagram: User Clicks on a Restaurant



B8. State Machine Diagrams

State machine diagram of Home Screen

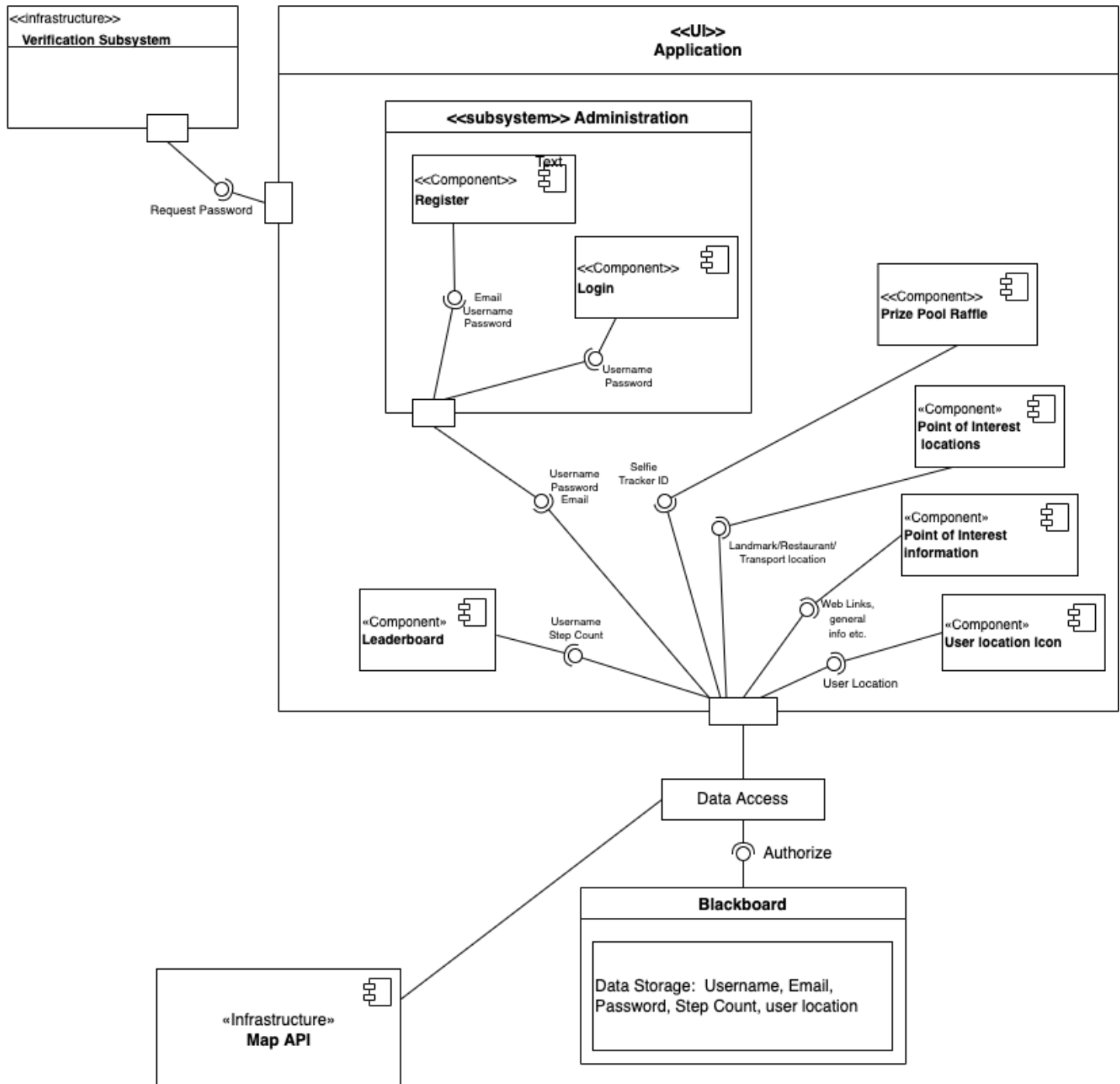


State machine diagram of the map for a registered user with location services on

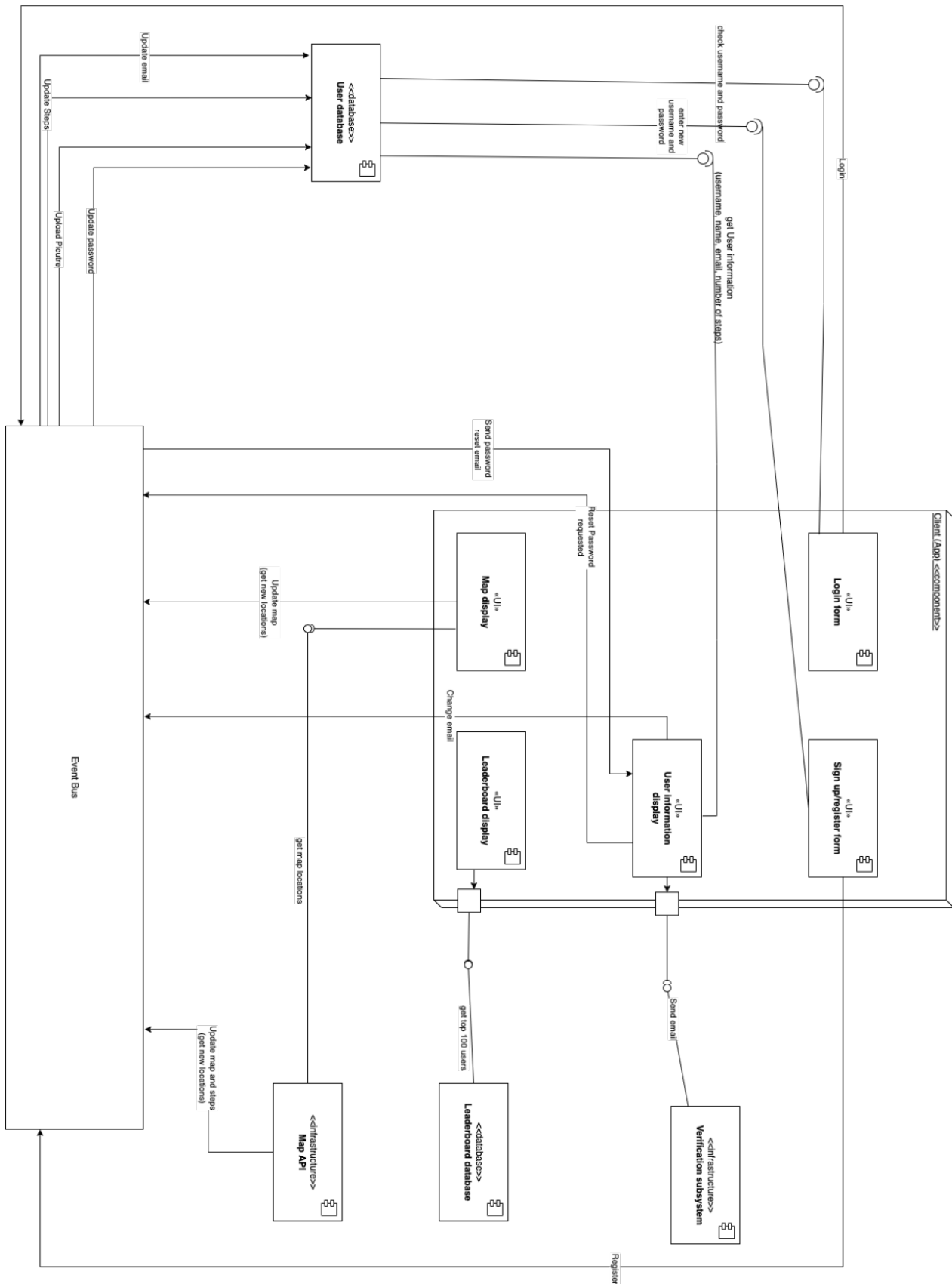
C. Software Architecture Style, Modelling and Evaluation:

C1. Component Diagrams

Blackboard Style

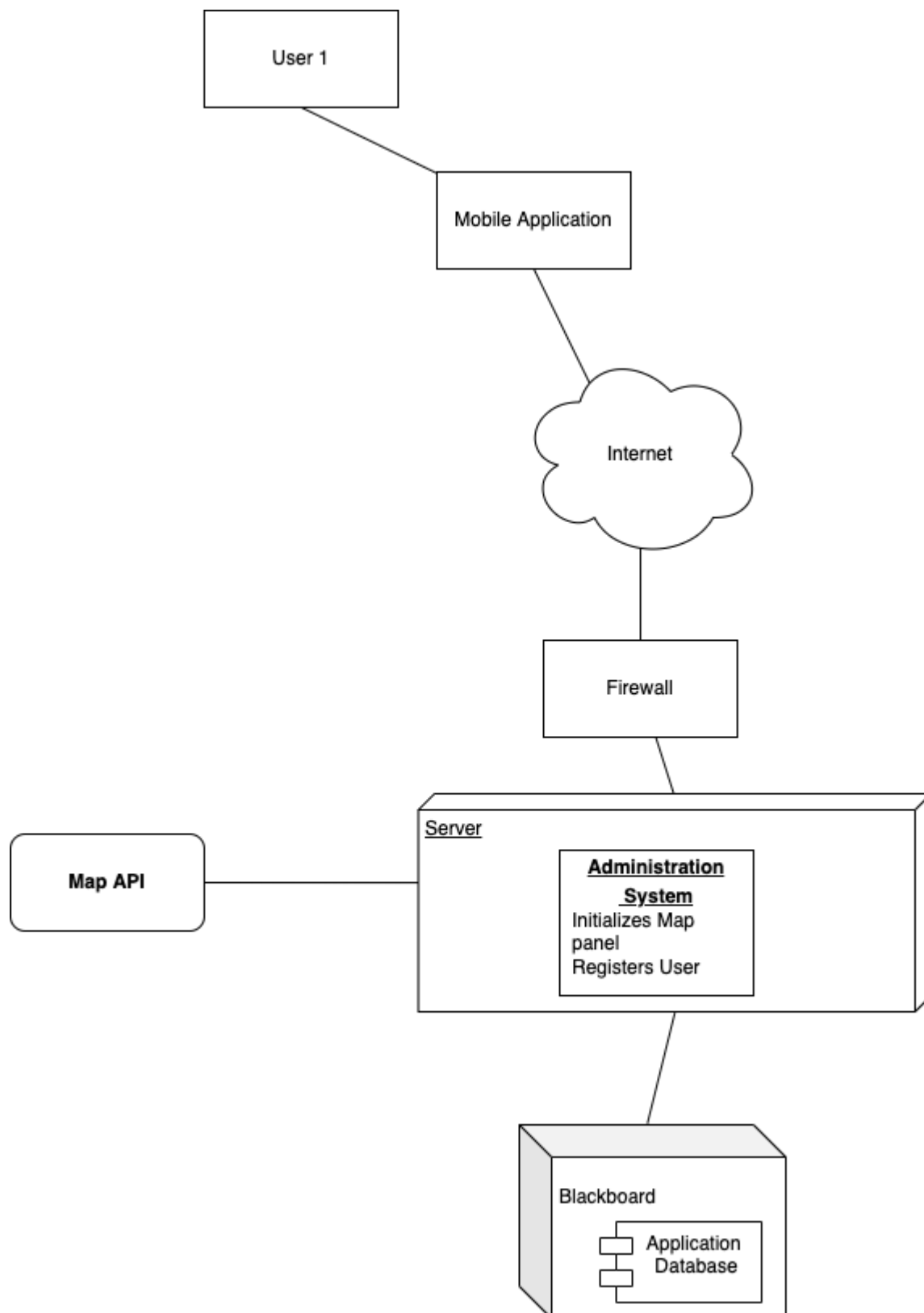


Event-Based Style

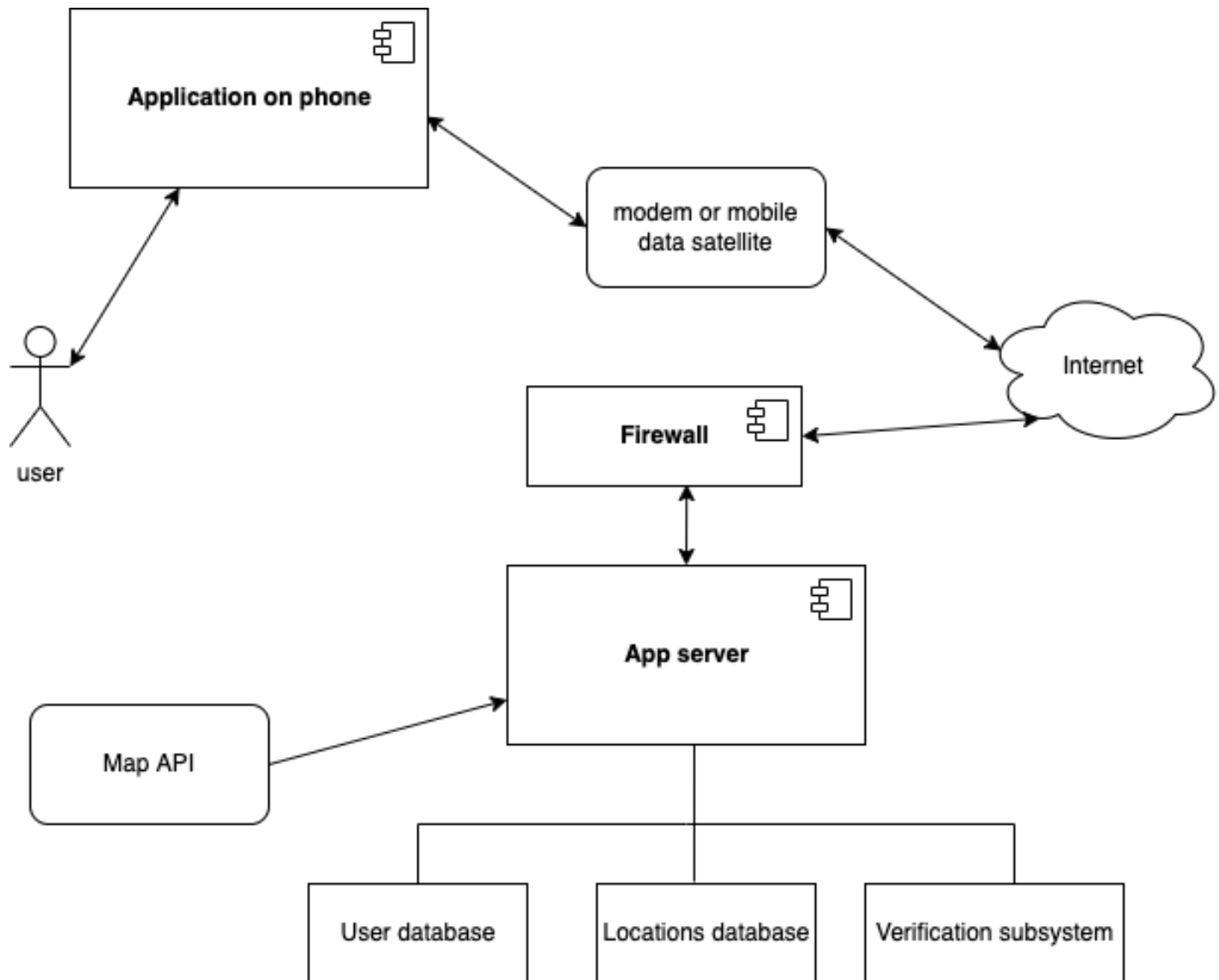


C2.Deployment Diagrams

Blackboard Style



Event-Based Style



C3. Architecture Comparison

One advantage the blackboard style has over event driven is that it is more efficient when sharing a large amount of data, because all the data is stored on one server whilst with event driven the data would be stored on multiple servers. Having the data on one server makes it easier to access as the system doesn't have to know which server to get the data from; it can just query the database on the server where the data resides. However, this makes data evolution more difficult compared to the event driven style. Having all the data on one server

means that evolving data could completely change how the data is structured in the database. This could cause a complete re-design of the database whilst with event driven architecture, we only need to worry about the effects on one database that we want to evolve which would be much cheaper in both time and money. For our app, evolution and maintainability is more important than sharing large amounts of data; if there are any changes to the venues or events of the commonwealth games, we need to quickly and easily reflect these changes.

One advantage that the event driven architecture has over the blackboard architecture is higher survivability due to the data being stored on multiple servers rather than one. For example, if one data server goes down due to a failure or attack then the whole system wouldn't necessarily go down, but rather just certain functionalities would become unavailable. If this were to happen with the blackboard style architecture the whole system would go down as all the data resides on one server, and this server is a point of failure. Furthermore, if one data server were breached then only the data on that server would become compromised, allowing us to take the necessary steps to protect the rest of the servers and mitigate or fix any damage done. Repairability and survivability is important to our system because the app must always be available during the commonwealth games, and we do not want a failure in one component to render the entire app unusable.

In conclusion, the event driven style architecture will be best as it provides a more robust way to store our data, allows easy maintainability and expansion of our app and protects us from failures caused both externally and internally.

D. Software Testing

Testing Objectives

The aim of our testing is to confirm the validation and verification of our system. We will use validation testing to check that the requirements have been met. We will also use defect testing to find any inputs that cause the app to behave incorrectly, so that we can then fix the fault that is causing the failure.

Testing Strategy

We will take a test-focused approach when developing the app. All tests will be written before the code to be tested is written. After implementing a single functionality, we will test it and continually refactor the code until it passes all tests. Once it passes the tests, we will move on to a new functionality.

For validation testing, we will provide a set of test cases that reflect the system's expected use. The app should respond as expected to these inputs and provide the required functionality. Our validation testing will be requirements-based; we will use the requirements to create our test cases. Our tests will show that our requirements have been satisfied.

For defect testing, we will provide test data that does not reflect the app's expected use. The system should be able to handle unexpected inputs. Our defect testing will use stress testing; the tests will be designed around the limits of the app and will make demands outside of the limits of our software. This will be used to make sure system failure does not cause data corruption or unexpected loss of services.

During validation and defect testing, we will use both black-box and white-box testing.

Towards the end of our system's development, we will also perform acceptance testing to verify that users are satisfied with our end product. The method used will be black-box testing, since we do not expect our users to

have an understanding of how the code works. This will test the system as a whole, rather than as individual components.

Testing Exit Criteria

Once 95% of all tests pass, we will move onto acceptance testing. Once our users approve the product, we will finish testing and deploy the app.

Test Cases for 7 Functional and 3 Non-Functional Requirements

Requirement ID	Test Case ID	Test description	Test steps	Test Data	Expected result
F1.2	01	Verify the login with valid email and password	Go to app login page Enter email Enter Password Tap 'Login' button	Email: testuser@gmail.com Password: testPassword	User should be able to login and access the features of the app. They should be taken to the Map page.
F1.2	02	Verify the login with valid email and invalid password	Go to app login page Enter email Enter Password Tap 'Login' button	Email: testuser@gmail.com Password: invalidPassword	User should not login into the application. The screen should display an "invalid password" error, allowing user to retry.
F1.2	03	Verify the login with an unexpected input	Go to app login page Enter email Enter Password Tap 'Login' button	Email: testusergmail.com Password: password	User should not log in to the application. The screen should display the appropriate error for the input given, in this instance "email with no @", and the user should retry logging in.
F1.1	04	Register with a new email	Go to Login page Tap 'register' button Enter email Enter password Enter username Tap 'OK' button	Email: newuser@gmail.com Password: newPassword Username: newUser	The user's details should be stored in the database and a new account should be created. The user should be taken to their profile page.
F1.5	05	Register with an email that	Go to Login page	Email: testuser@gmail.com Password: newPassword	No changes should be made to the database. The screen should display an "account

		is already in use	Tap 'register' button Enter email Enter password Tap 'OK' button		with that email already exists" error. The user should be taken back to the login page.
F2.1	06	Access the app as a guest	Go to Login page Tap 'continue as guest' button Go to the map section of the app.	Email: guest@guest	The user should be able to see and interact with the map.
F3.4.2	07	View information about a venue	Go to the map. Tap on the "Arena Birmingham" venue.	point_type = "venue" point_name: "Arena Birmingham"	The screen should display a pop up with the text: "Events at Arena Birmingham: Fri 29 July Gymnastics – Artistic Sat 30 July Gymnastics – Artistic Sun 31 July Gymnastics – Artistic Mon 1 August Gymnastics – Artistic Tue 2 August Gymnastics – Artistic Thu 4 August - Gymnastics - Rhythmic Fri 6 August - Gymnastics - Rhythmic Sat 7 August - Gymnastics - Rhythmic"
F3.4.5	08	View information about a landmark	Go to the map. Tap on the "Bullring" landmark.	point_type = "landmark" point_name: "Bullring"	The screen should display a pop up with the name of the landmark and the text: "The Bull Ring is a major shopping centre in central Birmingham. When combined with Grand Central it is the United Kingdom's largest city centre based shopping centre and has been

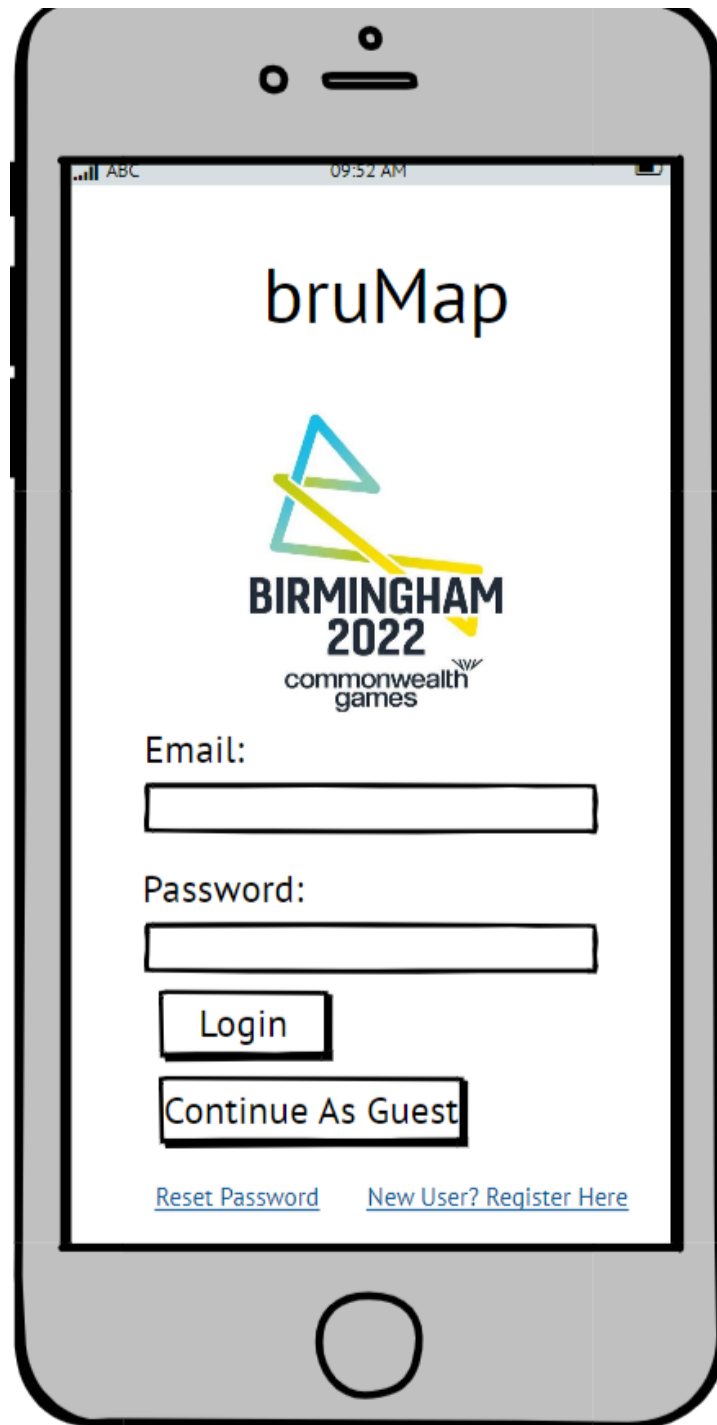
					an important feature of Birmingham since the Middle Ages, when its market was first held. “
F4.1.1	09	Unlock a fact by walking 1000 steps	Login to the app. Reach a step count of 1000. Go to the milestones page.	step_count = 1000	The “Unlocked Facts” section of the milestones page should show one fact randomly selected from the facts stored in the database.
NF1.2.1	10	Client’s response time to input is within 1 second	Login to the app. Press any on-screen input button, e.g. view information about the venue		The response time is within one second, in the instance of ‘view information about the venue’, the screen will display a popup stating the information within one second of input
NF1.2.3	11	The location updates every 2 seconds	Walk towards a place of interest, following the map		The user’s location will be updated on the map every 2 seconds.
NF3.1.1	12	In-app location is only shown if user provides their location access	Login to the app as user A, and provide location access. Log out. Login to the app as user B, and do not provide location access.		User A successfully logs in to the app, the app displays map with the user A’s current location. User A is successfully logged out. User B successfully logs in to the app, the app displays map without the user B’s current location.

E. Usability and Prototyping

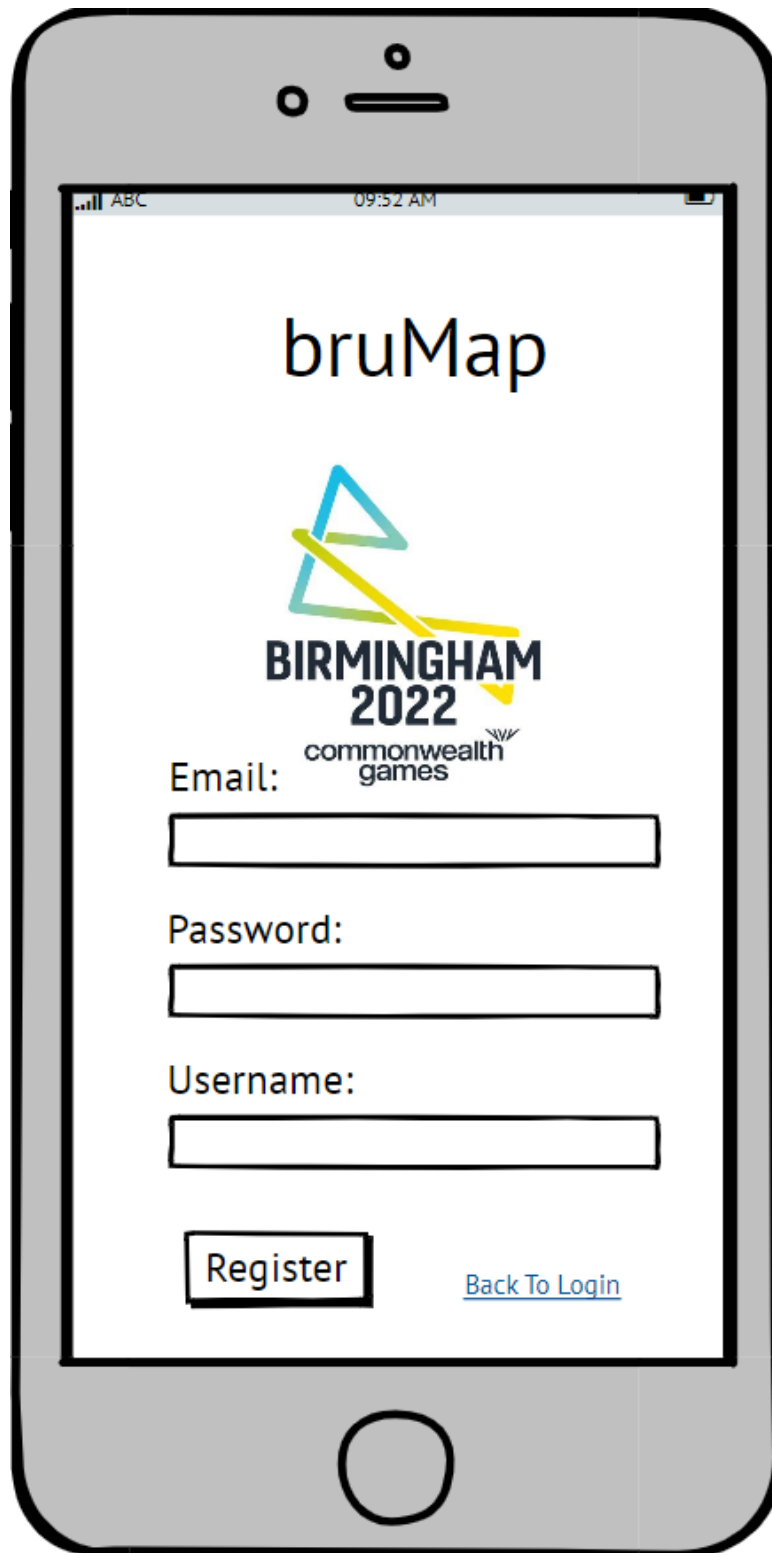
E1.

We used wire-framing to produce an interactive prototype showing the main structural details of the App. This prototype will be shown to stakeholders before we begin coding in order to ensure that the users are happy with the design and flow, and we don’t spend time coding things that the users do not want.

Login Page:



Register page:



A smartphone screen displaying the BruMap login and registration interface. The screen shows the 'bruMap' logo at the top, followed by the 'BIRMINGHAM 2022 commonwealth games' logo. Below the logos are three input fields for 'Email:', 'Password:', and 'Username:'. At the bottom, there is a 'Register' button and a '[Back To Login](#)' link. The status bar at the top of the phone shows 'ABC' and '09:52 AM'.

bruMap

BIRMINGHAM
2022
commonwealth
games

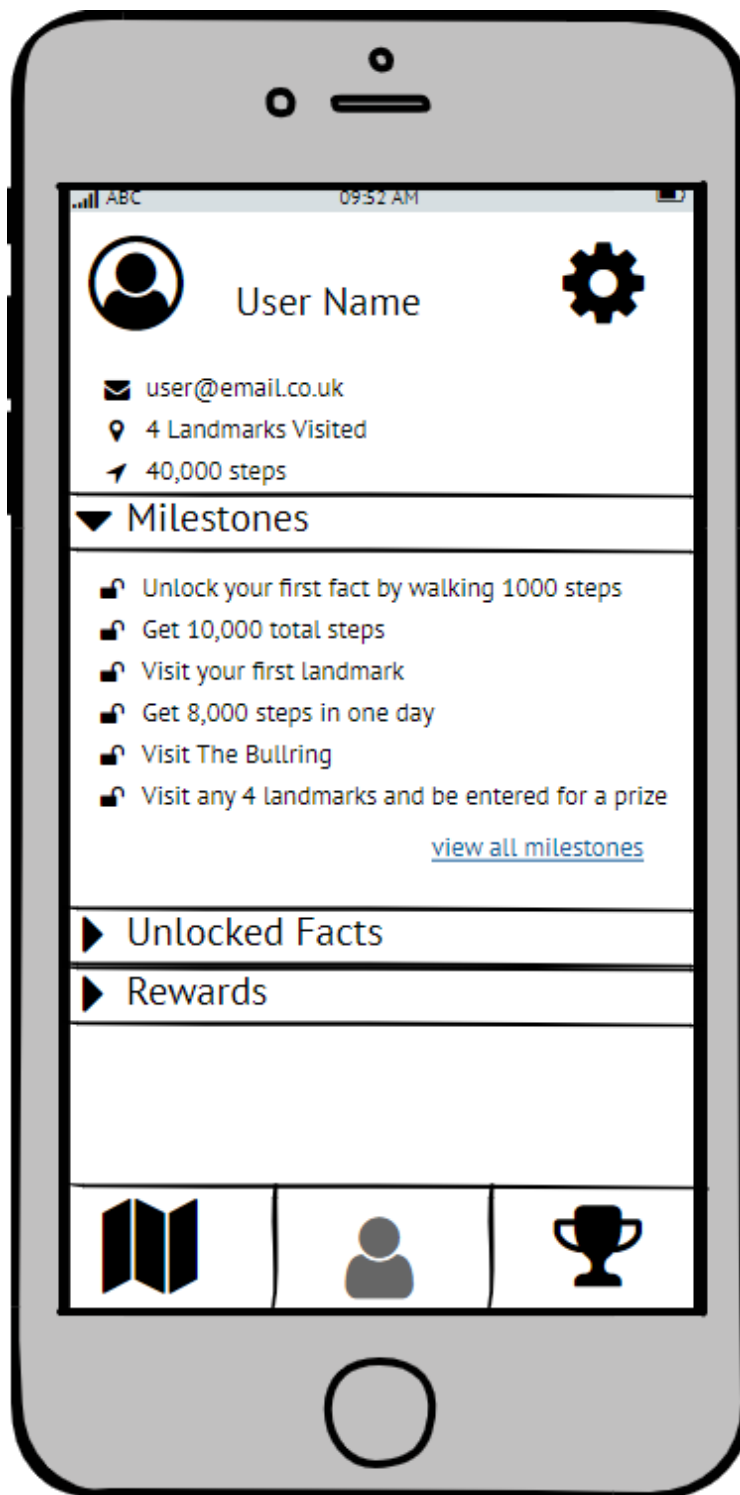
Email:

Password:

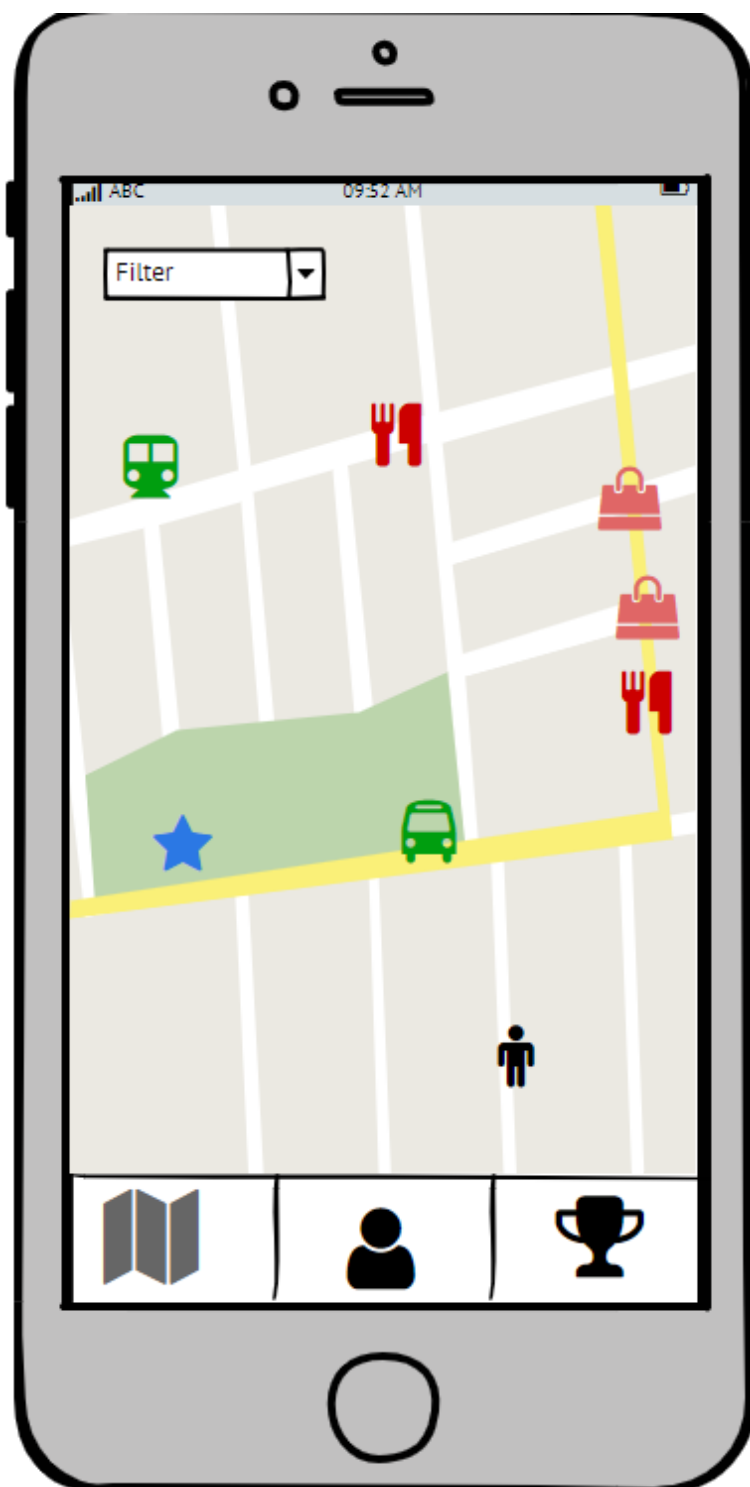
Username:

[Back To Login](#)

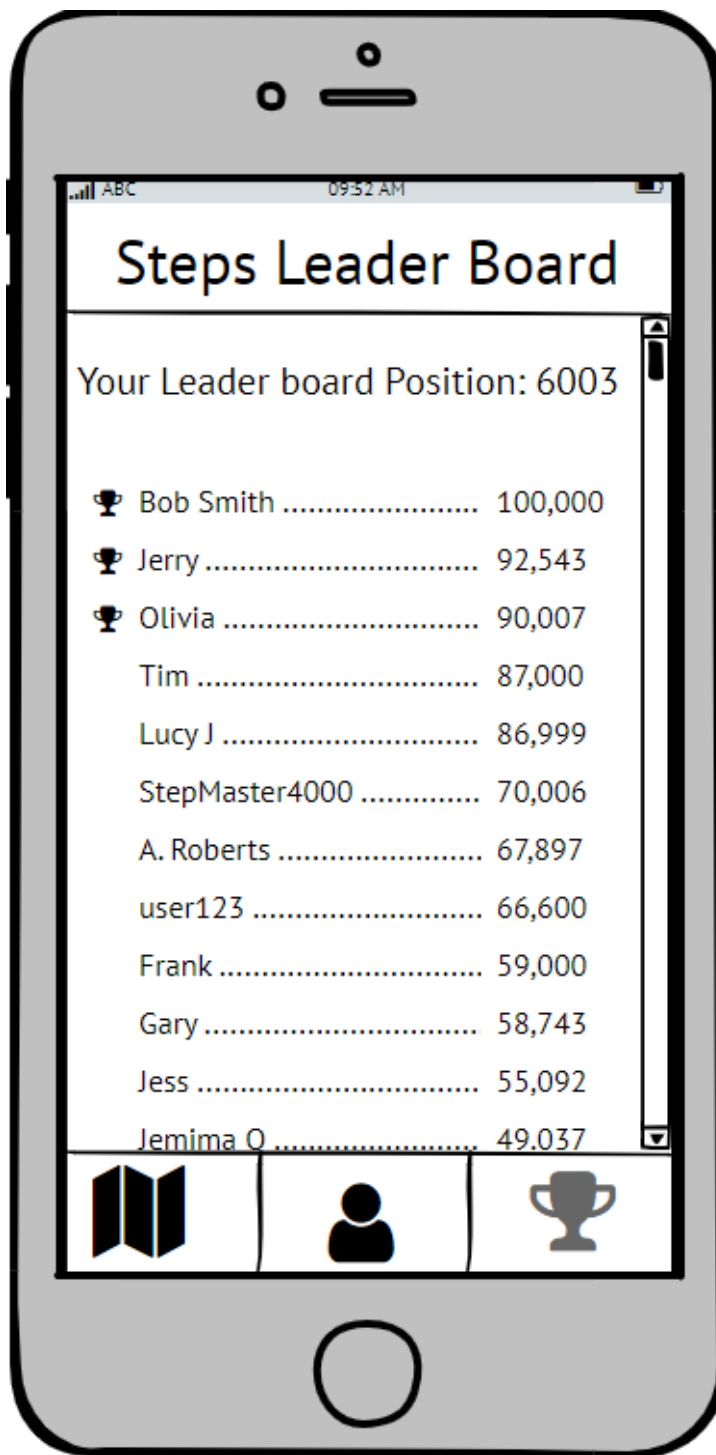
User Profile Page:



Map Page:



Leader board Page:



See included pdf document containing the interactive prototype.

E2. The video is contained within the files submitted.

F. Ethics and Professional Practice

In the design phase of this project, the ACM Code of Ethics was considered, and necessary alterations were made to meet the requirements. The interactive map design consists of ideas that will benefit the public, the visitors and the city of Birmingham. The application will be available for everyone, without an age limit. Certain accessibility issues will be addressed, such as using symbols and colours together to account for colour blindness. Furthermore, each of the images will have an alternative caption for people that are unable to see them. These implementations have been made with section 1.4 of the ACM of ethics in consideration. Once the user starts the app, they will be provided with the choice of creating an account or continuing as a guest. Data gathering will be minimized; the users will only need to share their email and location services to benefit from the milestones and rewards. The map will still be available even if the user chooses to continue as a guest; however, locations displayed may be less relevant to the user's current position. All users will be asked for consent before automatic data collection, and their data will be kept. The application will ensure the public is the central concern, as stated in section 1.1 of the ACM code of ethics. It will inform the users about the city of Birmingham and the Commonwealth Games by offering an interactive environment. The environment will promote restaurants, small shops and landmarks of Birmingham, engaging the visitors with the local economy. The ethical obligations and rules of the Birmingham City Council will also be adhered to. Finally, in accordance with section 2.5 and 2.9 of the ACM code of ethics, we have given comprehensive evaluation of the architecture of our system: specifically pertaining to the risks of having all the application data stored on a single server.