Requirements Specification

Project Name: *Castle*

Team Number 02

Document Information

Project Name:	Castle		
Prepared By:	Team 2	**Preparation Date:**	09/03/2022
Email / Phone:			
Document Version No:	6.0	**Document Version Date:**	10/05/2022

Version History

Ver. No.	Ver. Date	Revised By	Description
1	09.03.2022	Alex, Mai, Jing, Jiacheng	Created solution requirements
2	12.04.2022	Aysha	Executive summary, background & analysis
3	23.04.2022	Aysha	Edited and updated solution requirements
4	03.05.2022	Aysha	Edited and updated solution requirements
5	04.05.2022	Alex, Mai, Jiacheng	Added UML and ER diagrams, use cases and initial designs. Edited definition of terms
6	10.05.2022	Aysha, Jiacheng	Final edits

Table of Contents

0. Purpose (Executive summary)

We have been briefed to develop an app for university students that allows them to plan a day trip, using the local Newcastle transport services, to Alnwick Castle, Auckland Castle, Bamburgh Castle or Barnard Castle. The app should allow the user to purchase a ticket bundle, which includes same day return travel tickets and entrance tickets to the castle of their choice. The purpose of this document is to present the foundations of our work on this project, mapping out our journey from establishing the core requirements of the application to producing an initial design. We have approximately ten weeks to develop the application and so our primary goal is creating an app that is functional and fulfils the basic requirements given in the brief. Our secondary goals are to, firstly, ensure the app is attractive and intuitive to use. We believe this will be attainable within the timeframe as we can have the front-end and back-end production happen simultaneously.

---

1. Background & Analysis

1.1. Analysis Process

Our first steps were to break down what the core of the app-to-be was. Since the brief has asked for the app to help with travel to specific destinations, we decided it would be a combination of a transport and a travel guide app. We then began to form a vague idea of the structure of our app and the processes it would support, using other existing systems for guidance. Since we have used transport apps regularly, we are familiar with their domain and have used *Trainline* (thetrainline, 2022), *Google Maps* (Google LLC, 2022) and *Go North East* (The Go-Ahead Group PLC, 2022) for reference. In terms of travel guide apps, we are not as familiar, and so have looked to the travel guide apps *Wanderlog* (Travelchime Inc., 2022) and *GetYourGuide* (GetYourGuide, 2022) for reference.

1.2. Analysis

In terms of the environment for this domain, there are applications and websites that produce a route to the castles (*Google Maps*), sell tickets for those journeys (*Trainline*, *Go North East*) and provide information on the things to do when at the castles (the castle

websites[1]). Separately, these apps and websites support the processes required for our app however, our app will support all those processes in one place. The stakeholders for our app are University students who want to explore the castles surrounding Newcastle. We want the app to be intuitive for them to use and also make sure it appeals to a university student (e.g., no promoting attractions aimed for children).

1.3. Scope

The brief has indicated that this will be an app for university students to travel to specific castles from specific stations. That being known, below is a list of the functions we have explicitly agreed are in scope and out of scope:

The app *will*:

- Allow users to purchase tickets that provide them travel (from the specified stations) and access to one of the four castles mentioned in the brief and return to Newcastle in the same day.
- Present the user with additional information on what they can do around the castle.
- Provide the prices for student concession tickets.
- Present the user with information on the castles for them to make an informed decision on where to go.

The app *will not*:

- Allow users to travel between castles or start from any other station that is not Haymarket or Eldon Square.
- Allow users to purchase solely transport tickets or solely tickets to access the castles.
- Allow users to travel to a castle without a return or travel to a castle and return on a different day.
- Provide prices for anyone that is not a student.
- Allow users to book or purchase any extra amenities that surround the castle.

2. Hardware and software platforms to be used for developing and running your solution

- **Software platforms**

This project uses MySQL as the databased manager system via DBeaver. The server used in this app is based on Newcastle Computing school therefore a campus network is required in this app. JDBC is used to connect back-end code to the database.

The developing IDE in this project is Android Studio in the version 2021.1.1. The front-end part is implemented by Xml language and the back-end user interaction and the app structure are implemented by java in the version 11.0.

Github provided the synchronization and the version control of coding during the developing. Share Point is used for the online document shared.

- **Hardware**

Our app will be run on Android smartphones based on the system version from android 8.0 to android 12. The resolution ratio of the smartphone is from 1920×1080 to 3200×1440.

3. References

Alnwick Castle website: https://www.alnwickcastle.com/

Auckland Castle website: https://aucklandproject.org/

Bamburgh Castle website: https://www.bamburghcastle.com/

Barnard Castle website: https://www.english-heritage.org.uk/visit/places/barnard-castle/?utm_source=Trip%20Advisor&utm_campaign=Local%20Listings&utm_medium=Trip%20Advisor%20Profiles&utm_content=barnard%20castle&utm_source=Trip%20Advisor&utm_campaign=Local%20Listings&utm_medium=Trip%20Advisor%20Profiles&utm_content=barnard%20castle

GetYourGUide (2022), GetYourGuide: Tours & Tickets (version 3.94) [Mobile application software]. Retrieved https://apps.apple.com/gb/app/getyourguide-tours-tickets/id705079381

Google LLC (2022), Google Maps – Transit & Food (version 6.9) [Mobile application software]. Retrieved https://apps.apple.com/gb/app/google-maps-transit-food/id585027354

The Go-Ahead Group (2022), Go North East (Version 36.3.6) [Mobile application software]. Retrieved https://apps.apple.com/gb/app/go-north-east/id1556028079

thetrainline (2022), Trainline: Book train tickets (version 219) [Mobile application software]. Retrieved https://apps.apple.com/gb/app/trainline-book-train-tickets/id334235181

Travelchime Inc. (2022), Wanderlog (version 2.54) [Mobile application software]. Retrieved https://apps.apple.com/gb/app/wanderlog-travel-planner/id1476732439

4. Definition of terms

Some definitions are provided throughout the document

Activity: an application component that provides a screen with which users can interact to do something.

Adapter: an object that acts as a bridge between a listView and the underlying data for that view.

ListView: a configuration which could show a list of items, like the comments, tickets and the plans.

Android Studio: the official integrated development environment for Google's Android operating system.

API: an application programming interface, software intermediary that allows two applications to communicate with each other.

Class: a user defined blueprint or prototype from which objects are created in Java.

DAO: data access objects, used for operating the data from database.

Database: an organized collection of data stored and accessed electronically.

DBeaver: a SQL client software application and a database administration tool.

Dialog: a small window that prompts the user to select a choice or enter additional information.

End User: The person using the product

Fragment: An object which could represents a reusable portion of an app's UI.

Github: a provider of Internet hosting for software development and version control

HTTP: Hypertext Transfer Protocol, an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack.

IDE: An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development.

Java: an object-oriented programming language.

JDBC: Java database connectivity, the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems.

MySQL: an open-source relational database management system.

POI: places of interest, the places users might be interest around the castles.

UI: User interface, what the user sees

XML: a markup language used in the project to show the front-end configuration.

5. Solution requirements

5.1. Functional Requirements

	Requirement	Priority	Supplier Comments
Registration			
FR1	End user can register a new account with a username, an email address, and a password which should be entered twice	M	
FR2	If the end user enters a username and/or email that has already been used to register, a message will be shown saying their username/email has already been used	M	
FR3	If the two passwords entered by the user do not match, a message will alert them that they have entered different passwords	M	
FR4	If the end user enters a non-university email address, a message will alert them to use their university email address	M	For simplicity we have said university emails will be of the format [StudentUsername]@[UniversityName].ac.uk
Log In			
FR5	End user can log into their account using their email/username and password	M	
FR6	If the end user enters the incorrect username or password, a message will alert them	M	
FR7	End user can click “forgot password” to view their password by providing the correct email and username combination	M	
FR8	Usernames, passwords, and emails will be stored in a database securely	M	
Side Navigation Bar			
Account details			
FR9	End user can change their profile picture using their device’s camera or by exporting a photo from their photo library	L	
FR10	End user can change their account details (email address, nickname, password, and bank details)	L	

FR11	End user cannot change their username	L	
FR12	The database will update accordingly with the end user's changes	H	
	Settings		
FR13	End user can change the default language the app is set to (English or Chinese)	L	
	Reviews		
FR14	End user can post reviews and ratings of the application	L	
FR15	End user can view other users reviews and ratings of the app	L	
	Log Out		
FR16	End user can log out of their account	M	
	Bottom Navigation Bar		
	Castles		
FR17	End user can select a castle and view information on that castle	H	Information on the castles includes images, a description, prices, location, opening hours and close by amenities
FR18	End user can get further information on the castle by clicking on their website links	M	
FR19	Information on castles will be retrieved from a database	H	
	Booking		
FR20	End user can select • The castle they wish to travel to (Alnwick, Auckland, Bamburgh, Barnard) • The date they would like to go • The departure and return time • How many people they wish to purchase a bundle for (up to 5)	H	
FR21	End user should not be able to book tickets for times outside of the bus/train timetables nor outside of the castle opening times or in the past	H	
FR22	After selecting their travel preferences, the system should present the user with a choice of routes and forms of travel	H	

	they can take to the castle from Newcastle City Centre		
FR23	End user can select then pay for or save their chosen ticket bundle	H	*Ticket bundle*: the tickets for travel and entrance to (and from) the castle
FR24	Information on travel will be retrieved from a database and inputted manually	H	
My Plans			
FR25	End user can view their upcoming plans. This will show the date and time of travel, the number of tickets and whether they have paid or not	H	
FR26	End user can view details of specific plans. This will be the details of the journey to and from the castle and nearby amenities	M	
FR27	End user can refund some or all their ticket bundles	M	
FR28	End user can pay for their ticket bundles if they haven't already done so	H	
Team info			
FR29	End user can view information on and contact details of the development team of the app	L	
General			
FR30	End user can return to the home page from any screen on the app (after they've logged in)	M	The home page being the castle display page
FR31	Layout of the app will match up with phone orientation	L	

5.2. Non-Functional Requirements

	Requirement	Priority	Supplier Comments
NFR1	For security passwords will be encrypted within the database	M	This partially fulfils FR8
NFR2	End user will be required to be connected to Newcastle University or Eduroam wi-fi for the databases to be accessed	H	This partially fulfils FR8, FR12, FR18, and FR23
NFR3	FR9 may require a phone with a working camera	L	This fulfils FR9

NFR4	Reviews will be moderated for foul language	L	This partially fulfils FR14
NFR5	Reviews will have a subtitle of maximum 30 characters, a body of maximum 600 characters, and a rating of maximum 5 stars	L	This partially fulfils FR14
NFR6	Information on the castles will be gathered from their websites	M	This fulfils FR17 and FR19
NFR7	The process of selecting travel preferences should take 12 clicks/taps if done with certainty	M	*Certainty*: Each selection is the final and intended selection of the end user This fulfils FR20
NFR8	The process of selecting and paying for/saving a bundle should take no more than 3 clicks if done with certainty	M	This fulfils FR23
NFR9	Real life data from travel services will be used	H	This fulfils FR24
NFR10	Fonts used in the GUI will be legible. Sans serifed, well-spaced, and match system font size settings	M	
NFR11	GUI will be simple. Not busy with unnecessary images, icons, patterns, or information	M	
NFR12	Error messages will be in plain English or Chinese, dependent on language settings	M	This fulfils FR2, FR3, FR4, FR6, FR11, and FR21

6. Other considerations

6.1. Assumptions

- The application is aimed for university students, so we have assumed using a smartphone application will be intuitive. By intuitive, we mean they do not need hints/directions in the use of the app.
- Customers will be at and/or around the castle for a minimum of two hours and so outbound travel tickets will be available for the morning
- End user's devices are in the GMT/BST time zone
- Tickets will not sell out for the castles nor for the buses and trains
- Bus and train prices are fixed
-

6.2. Constraints

- We have been given approximately ten weeks to develop the application
- For some of us, app development is a completely new activity

- The app is fairly local, meaning it does not access any external sources of information (i.e. people's social media accounts, google maps, direct information from castle websites)

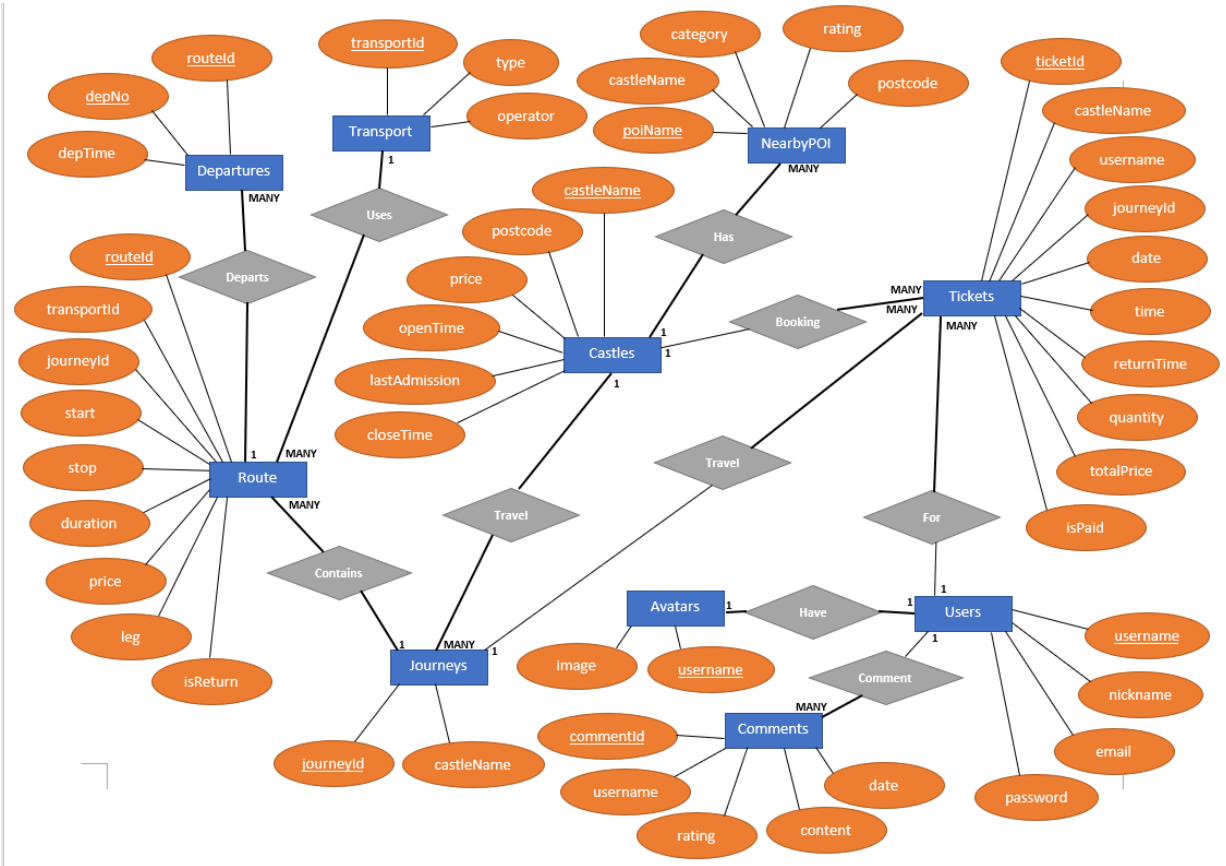
6.3. Dependencies

- Our app will be made specifically for Android
- Access to the database on DBeaver requires the end-user to be connected to university wi-fi in order for the databases to be updated

7. Initial requirements analysis of user interaction

7.1 UML and ER diagrams

1. ER-diagram for the database

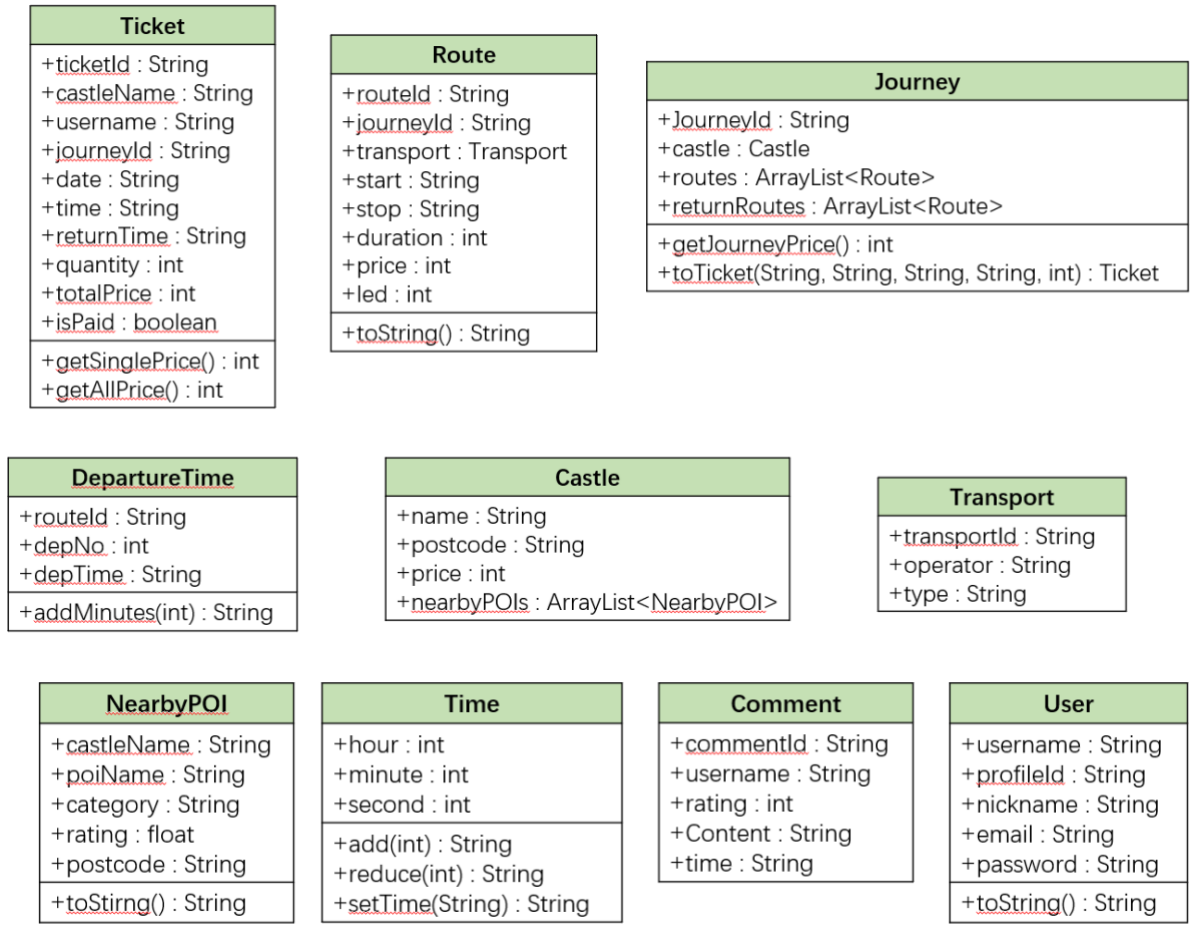


Assumptions

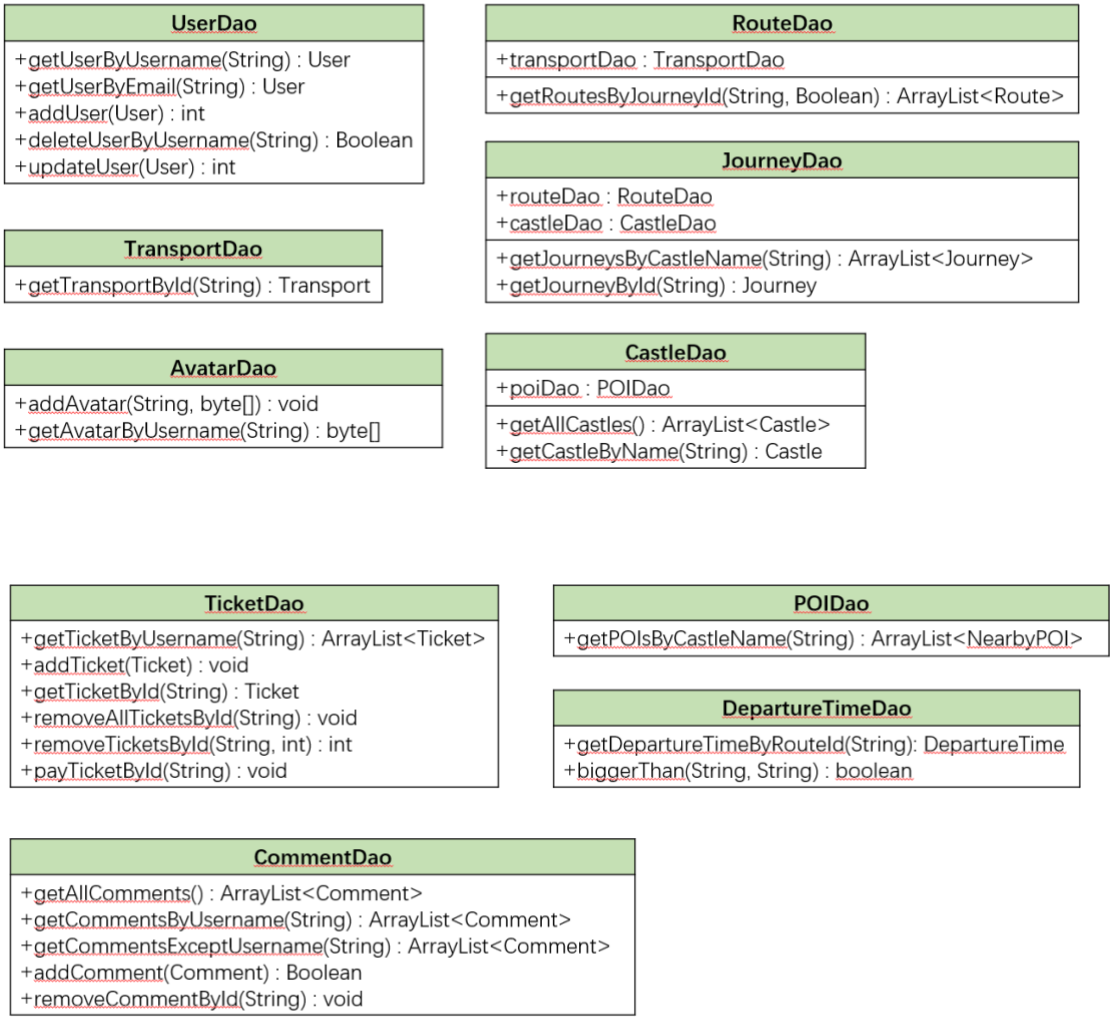
- Tickets are created once they are ordered.
- Every user must have an avatar.

Note: The Avatar table only exists separately because loading the user's avatar for every action that uses the 'Users' table would cause the app to run slower.

2. Class Diagram

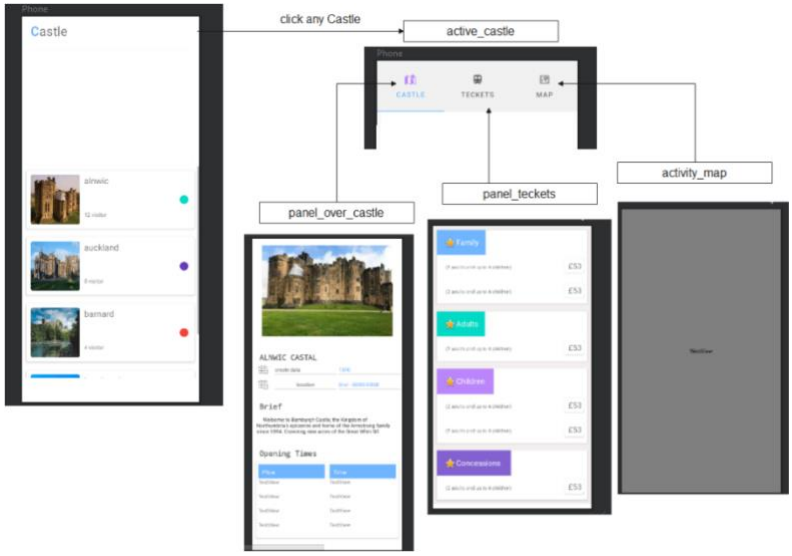


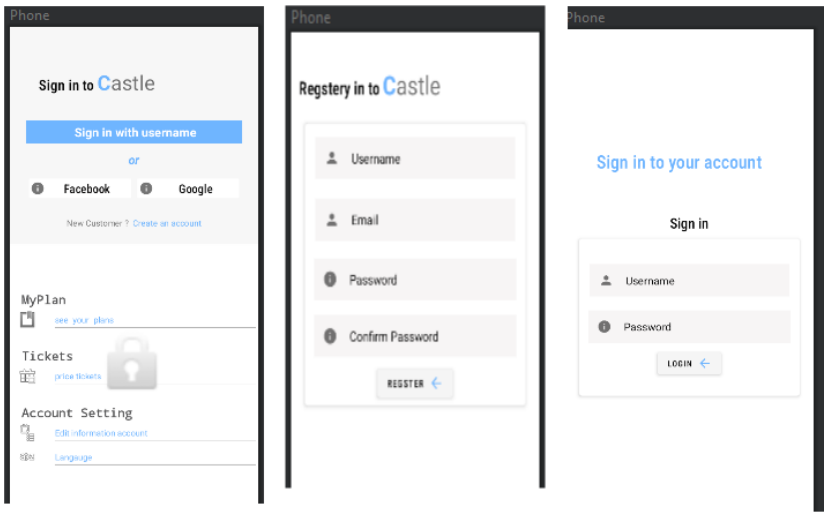
Daos: responsible for CRUD to every table in database



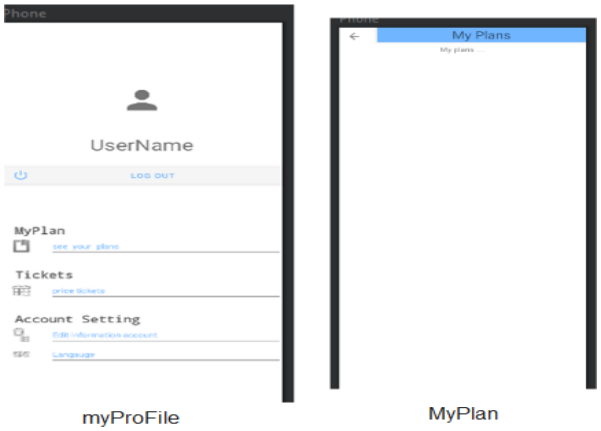
7.2 Initial ideas for a GUI

Using what we have learnt from Trainline and GoNorthEast, we have made the following initial designs of the application:





myProFile sign up log in

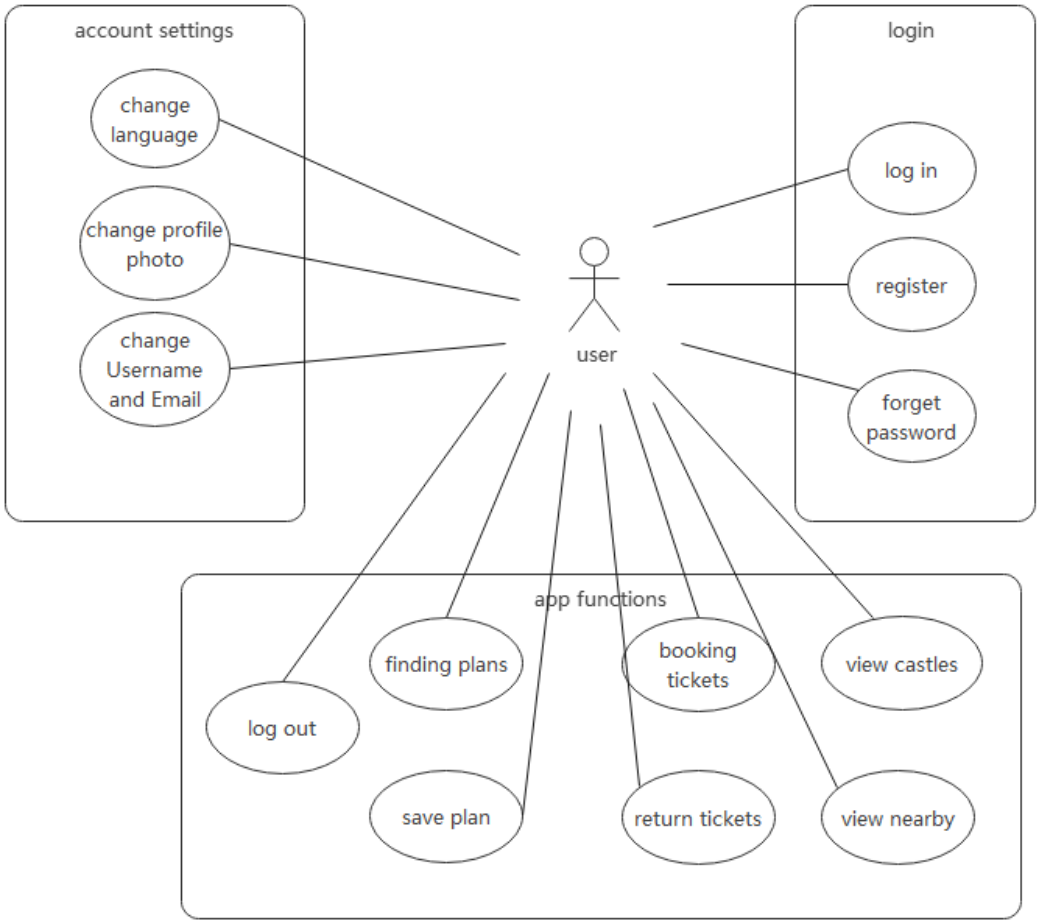


myProFile MyPlan

We noticed it was common for applications to have a bottom and/or side navigation panel which we have made use of. We think the core of this design will remain the same with some minor changes as we move throughout the project.

7.3 Use cases

Use case diagram:



User accesses:

a. Login part

1. register to the system as a new user

User action	System responses
1. User presses a button in the login page to register 2. User enters a valid university email address, a new username, and password.	1. The system will send the user to a register page 2. The system will register the user to the account database and then send the user to the main login page

2. Log in to the app

User action	System responses
1. User presses a button to log in 2. User enters a wrong account 3. User enters a wrong password 4. User enters a correct password	1. The app will send the user to the log in page 2. The system will inform the user that their account doesn't exist 3. The system will check the password in the database and return a message that the entered passwords are mismatched. 4. 4. The app will send the user to the main page and record their account

3. forgot password

User action	System responses
1. User presses a button in the login page to say they have forgotten their password 2. User enters their email, and the associated username then presses the find password button	1. The app will take the user to the forgot password page. 2. The system will check the email and username to see if they are valid and then provide the user with their password

b. Account settings part

1. Change language

User action	System responses
1. User opens the account details bar by pressing the profile photo 2. User presses change language 3. User selects their preferred language	1. An account bar is displayed which contains all of the account functions 2. A new page is displayed 3. The app returns to the main page with the language changed

2. Change profile head photo

User action	System responses
1. User presses the head photo at the account bar 2. User presses the detail photo 3. User chooses a photo from album or camera. Then presses save.	1. A larger photo detail will display 2. Two options would display, take a photo or choose from album 3. The avatar photo changes to the selected/taken photo

3. Change username and email address

User action	System responses
1. User presses the options in the account bar to enter the username changed 2. User enters a username which is used by another person 3. User enters a valid username	1. The app would turn to a page for changing the username 2. The database checks the username and informs the user to select another username 3. The system would change the account information in the database and change the information displayed in the app.

c. App main function part

1. Find plans

User action	System responses
1. User presses the find button in the bottom navigation bar 2. User selects a castle and selects a date/time to find plans	1. The app would display the find plans fragment 2. The app would go to a page and show the results of all the suitable plans

2. Save plans

User action	System responses
1. In the find plans result page, user selects a preferred plan. 2. The user finds this plan and if it is suitable they press the save button to save this plan	1. The details about this plan would display 2. This plan would save into this account database. And it would display in my plan page.

3. Booking tickets

User action	System responses
1. User goes to my plans fragment by press the button My plans in the bottom menu 2. User selects a saved plan 3. User presses buy ticket button	1. The app will go to my plan page 2. The plan details shown in this page 3. The system will buy a ticket(s) and save the ticket to the database.

4. Return tickets

User action	System responses
1. User presses a ticket that has been bought 2. User presses the refund button 3. User chooses a number and press refund in the dialog	1. The ticket info is displayed 2. The system will display a dialog to make sure the user wants to refund the ticket and to get the number of ticket user wants to refund 3. The system refunds the ticket and returns the money to the user account. The ticket is removed from the database.

5. View castles

User action	System responses
1. User presses the castle button in the main page bottom menu 2. Presses a castle view	1. The app will display the page for the four castles 2. App goes to a page that displays the castle info. This page includes the information like the price of the castle, the description of the castle.

6. View nearby

User action	System responses
1. User presses the nearby button in the castle info page	1. The app will display the nearby places including shops and cafes.

7. Log out

User action	System responses
1. User presses the account header photo to enter the account bar 2. Presses the logout button	1. The account bar is shown in the main page 2. The user logs out and the app sends them to the main log in page