TRIP TRACK

REQUIREMENT ANALYSIS

Navinder Waraich, Natalia Witkowska, Naomi Durkin, Jason Lika, Benjamin Cuello-Wolffe, Conlan Roberts

PROJECT DESCRIPTION

Trip Track is a mobile application designed to help travellers plan and document their travel experiences. The app provides a way for users to log their travel activities, while also allowing them to capture their memories through photos and journal entries. The app aims to provide a user-friendly interface and features that make it easy to capture travel experiences and keep them organised. The app is going to be developed using Swift.

AIMS & OBJECTIVES

AIMS:

- To create an easy-to-use platform that is a central hub for travels, including organising and documenting their trips
- To provide users with a space to plan future travels and also review their past travels.
- To provide users with an intuitive and user-friendly interface that makes it easy to navigate the app and create new trip plan and journal entries.

OBJECTIVES:

- To ensure the app is compatible with iOS.
- To allow users to easily create new journal entries for specific locations on the map or their current location.
- To let users add details to their journal entries like the date visited and their notes for that location and allow them to attach photos of the location.
- To show the users visually what locations they have visited.
- To allow users to view and edit all their journal entries.
- To allow users to easily create new trip plans so they can keep all details in one place.
- To allow users to add details to their trip plans like: city, date going/leaving, transportation methods and details, accommodation type and details, packing list and activities list.
- To offer users the ability to categorise their trip plans into three sections: planning, upcoming and completed and make it evident which category the trip is in.
- To send the users notification reminders when their upcoming trip is approaching.
- To make sure the users are informed about what data we are collecting from them.
- To make sure the users are informed about their ethical and legal responsibilities.

KEY LITERATURE & BACKGROUND READING

Mobile applications are application software that are designed to be used on **mobile devices** like a phone or tablet rather than on a computer, this must be kept in mind when designing the UI and other aspects of the app. The Apple App Store has over 3.5 million apps itself [1].

When designing the user interface, we followed the **Human Interface Guidelines** [2] that have been written out by Apple which contains guidance and best practices to use when designing a system to ensure the user has a good experience when using your application. These guidelines include things like asking for user data and input, using globally recognised icons and other intuitive design features which were all incorporated into the UI mockup. Background reading was also carried out in the testing section and in the planning section.

DEVELOPMENT & IMPLEMENTATION SUMMARY

We plan to make this app in **Swift**, the native iOS application coding language which ensures the app is compatible with iOS.

We'll be coding using **XCode** as it is the IDE that all team members have experience using. We chose this as the majority of the team are confident with developing using Swift and all of us find it enjoyable to use. We also plan to utilise GitHub to share code and files. All user data will be stored in the app's local **Core Data**.

The app will offer the user three main views through a **toolbar: map view, journal view, and plan view**. The main view is the map view, which has a visual representation of the places they've logged as visits (which is one of our objectives). You can click on the map to add an annotation representing a location visited or click on the '+' icon in the top right to add an entry at the user's current location. Clicking on a previously added annotation will open that entry's journal details.

Adding a new entry opens a form of text and photo fields. Once these have been adequately filled, the user exits the entry and it's immediately attached to the appropriate location. **Journal entries** can be edited by clicking the 'edit' button in the navigation bar while viewing. This journal entry will begin as a page of text but has the potential to evolve with additions like music that plays when opened or photos and stickers which can be placed on the entry's page.

Adding a new plan opens a form including text fields like 'City' and 'Start Date.' Adding a new plan appends it to the table in the plan view. The plan view offers the user the easy ability to create new detailed plans for upcoming trips. All the features listed above have been thought of by going over objectives together and coming up with features that can solve them.

DATA SOURCES

- **Mapping Data:** The app displays the map using Swift's MapKit framework.
- **Location Data:** The app will use location data to determine the user's current location. With this data, the user's location is displayed on the map and so the user can add their journal entries to their location. This will be obtained with the user's knowledge and permission.
- **Local Laws and Sensitivities:** The app will rely on third-party sources, such as the GDPR website, to provide necessary information about local laws and regulations of the countries in order for the app and users to comply with these.
- **Cities Data:** On the planning screen the user will be able to choose the city they want to go to from a dropdown. The list of cities for this dropdown will be taken from a publically available list of UN recognised cities from the Internet.

TESTING & EVALUATION

We will test our code **against our aims and objectives** to ensure that we are making our application according to the specification we set out. We will ensure that we leave adequate time for testing during the planning of our application. **Validation** will be included in all the setters, so validation is only performed once. This will ensure that code can be tested easier.

In order to test our application, we plan to utilise **black box testing**, such as orthogonal array testing. We will also use **whitebox testing** in order to test our application. This will include **path testing** to ensure that every line of our code has been tested. We will conduct **usability testing** by getting at least three volunteers to use our application. We will make sure to do this ethically and abide by the university's policy on research ethics. We will use **laboratory usability testing**, so we will need an observer to watch the volunteer use our system in silence and take notes on their behaviours and report the outcome of the testing.

We will use the **XCTesting framework** in order to write our unit tests. This will allow us to automate the testing and allow us to validate the code in the unit we're testing to ensure it behaves as intended. Each of our tests written using the testing framework will be named descriptively and concisely in order to make it clear what each of our tests are testing for. The XCTesting framework will also be used for the performance testing of our application as it has options that allow for the gathering of performance measurements.[3] We will record all our tests in a **table** along with the expected and actual results.

ETHICAL CONSIDERATIONS

All members have read the university's ethical guidelines and will follow them throughout the duration of this project.

Data Security: The app collects personal information about its users, including their travel history and future plans. It is therefore important that the app has security measures in place and will allow users to have control over their data and decide who can see it. Security measures can include encryption etc.

Accessibility: The app should be accessible to a range of people including those with disabilities and so the design should be inclusive and user-friendly. For example, those who are colour blind can distinguish between two different icons using their shapes instead of just their colour.

Fair Use of Data: The app should not use the user's data for purposes beyond what was agreed to by the user themselves. This includes using data for targeted advertising or selling it to third parties.

Respect for Local Laws and Sensitivities: The app should comply with local laws and regulations of the countries it displays including those related to the use of geolocation data, data protection and privacy.

Trip Planning and Safety Responsibilities: While the app can be a useful tool for planning trips, users must remember that it is their own responsibility to keep themselves safe when travelling. Other sources such as travel advisors and government websites can be used to view information on the potential dangers and risks that are associated with their travel plans.

Location Data: In order to allow the app to use location data to show the user's travel destinations on a map, the user must provide consent. It's vital to be clear about how this data will be used and allow the users to have full control over their location data, such as being able to turn off the location services or delete previous location data. The app should also ensure that the location data is collected and stored securely.

Photo Usage: The app should receive the user's consent before being able to access and use their photos. The user may also wish to allow the app to access only certain photos from their gallery through their selection. These photos must be stored securely and not shared without the user's permission.

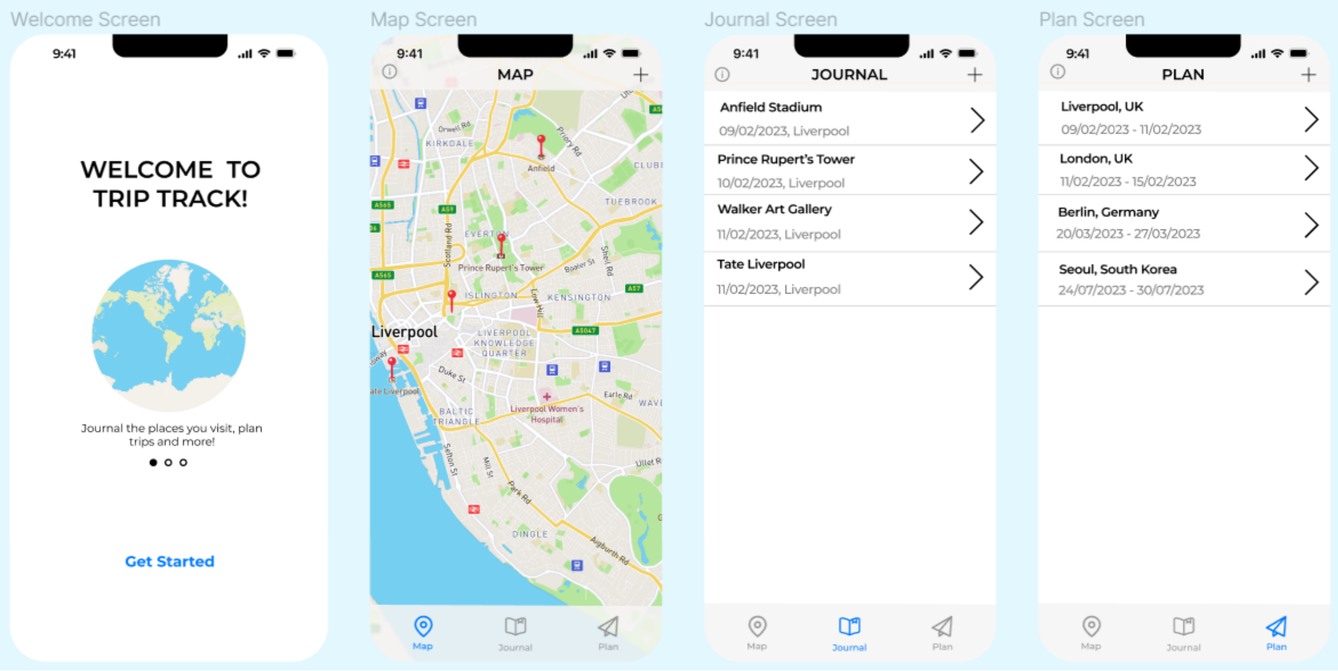
BCS PROJECT CRITERIA

We will be applying analytical skills we learnt in **COMP201** and **COMP107** such as requirements analysis, testing and other skills needed for group work. We'll also be applying practical skills we learnt in **COMP228** (programming using Swift).

We have successfully used innovation and creativity in order to create an app idea which combines two major areas of mobile applications (**journaling and travel**) and is feasible to produce within the given time frame.

In order to prove our ability to self-manage this project, we have created a **Gantt chart** to show we can allocate time and tasks efficiently and we also have **regular meetings**, which also helps with synthesis of information and ideas, so all group members are on the same page. We also evaluate ourselves and other team members by **providing feedback and improvements** during these meetings.

UI/UX MOCKUP



Welcome Screen

WELCOME TO TRIP TRACK!

Journal the places you visit, plan trips and more!

Map Screen

Journal Screen

JOURNAL

Anfield Stadium
09/02/2023, Liverpool

Prince Rupert's Tower
10/02/2023, Liverpool

Walker Art Gallery
11/02/2023, Liverpool

Tate Liverpool
11/02/2023, Liverpool

Plan Screen

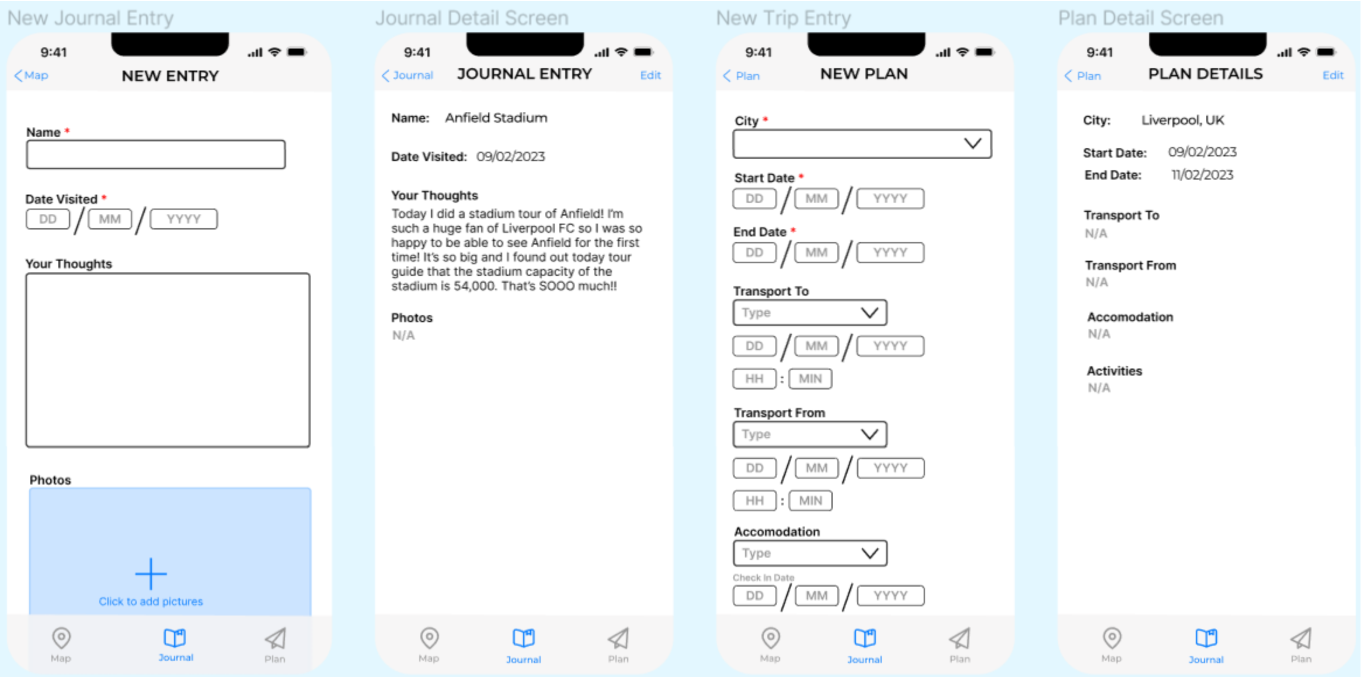
PLAN

Liverpool, UK
09/02/2023 - 11/02/2023

London, UK
11/02/2023 - 15/02/2023

Berlin, Germany
20/03/2023 - 27/03/2023

Seoul, South Korea
24/07/2023 - 30/07/2023



New Journal Entry

9:41

< Map NEW ENTRY

Name *

Date Visited * DD / MM / YYYY

Your Thoughts

Photos

Click to add pictures

Journal Detail Screen

9:41

< Journal JOURNAL ENTRY Edit

Name: Anfield Stadium

Date Visited: 09/02/2023

Your Thoughts

Today I did a stadium tour of Anfield! I'm such a huge fan of Liverpool FC so I was so happy to be able to see Anfield for the first time! It's so big and I found out today tour guide that the stadium capacity of the stadium is 54,000. That's SOOO much!!

Photos

N/A

New Trip Entry

9:41

< Plan NEW PLAN

City *

Start Date * DD / MM / YYYY

End Date * DD / MM / YYYY

Transport To

Type

DD / MM / YYYY

HH : MIN

Transport From

Type

DD / MM / YYYY

HH : MIN

Accomodation

Type

Check In Date

DD / MM / YYYY

Plan Detail Screen

9:41

< Plan PLAN DETAILS Edit

City: Liverpool, UK

Start Date: 09/02/2023

End Date: 11/02/2023

Transport To

N/A

Transport From

N/A

Accomodation

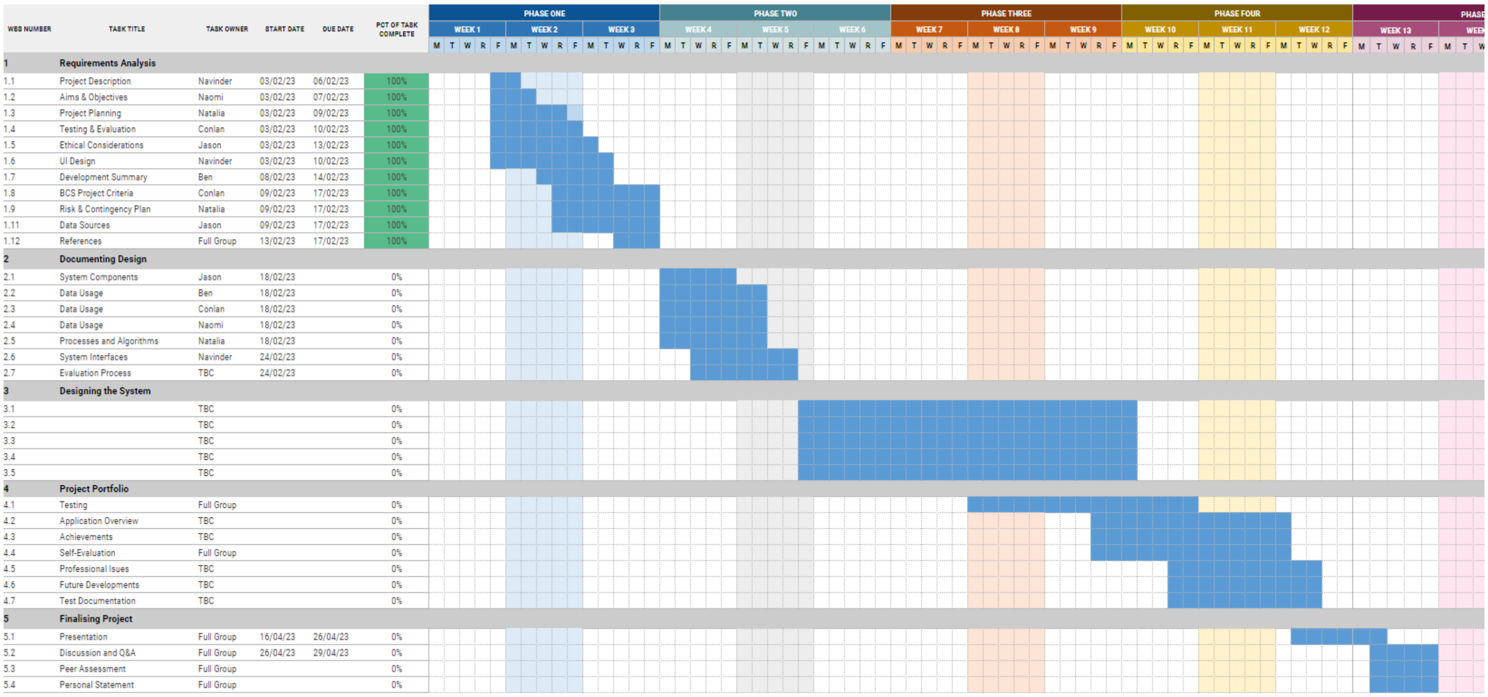
N/A

Activities

N/A

PROJECT PLAN, RISKS & CONTINGENCIES PLAN

PROJECT MEMBERS	Ben Cuello-Wolffe, Conlan Roberts, Jason Lika, Naomi Durkin, Natalia Witkowska, Navinder Waraich	END DATE	12/05/23



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	PCT OF TASK COMPLETE
1	**Requirements Analysis**				
1.1	Project Description	Navinder	03/02/23	06/02/23	100%
1.2	Aims & Objectives	Naomi	03/02/23	07/02/23	100%
1.3	Project Planning	Natalia	03/02/23	09/02/23	100%
1.4	Testing & Evaluation	Conlan	03/02/23	10/02/23	100%
1.5	Ethical Considerations	Jason	03/02/23	13/02/23	100%
1.6	UI Design	Navinder	03/02/23	10/02/23	100%
1.7	Development Summary	Ben	08/02/23	14/02/23	100%
1.8	BCS Project Criteria	Conlan	09/02/23	17/02/23	100%
1.9	Risk & Contingency Plan	Natalia	09/02/23	17/02/23	100%
1.11	Data Sources	Jason	09/02/23	17/02/23	100%
1.12	References	Full Group	13/02/23	17/02/23	100%
2	**Documenting Design**				
2.1	System Components	Jason	18/02/23		0%
2.2	Data Usage	Ben	18/02/23		0%
2.3	Data Usage	Conlan	18/02/23		0%
2.4	Data Usage	Naomi	18/02/23		0%
2.5	Processes and Algorithms	Natalia	18/02/23		0%
2.6	System Interfaces	Navinder	24/02/23		0%
2.7	Evaluation Process	TBC	24/02/23		0%
3	**Designing the System**				
3.1		TBC			0%
3.2		TBC			0%
3.3		TBC			0%
3.4		TBC			0%
3.5		TBC			0%
4	**Project Portfolio**				
4.1	Testing	Full Group			0%
4.2	Application Overview	TBC			0%
4.3	Achievements	TBC			0%
4.4	Self-Evaluation	Full Group			0%
4.5	Professional Isues	TBC			0%
4.6	Future Developments	TBC			0%
4.7	Test Documentation	TBC			0%
5	**Finalising Project**				
5.1	Presentation	Full Group	16/04/23	26/04/23	0%
5.2	Discussion and Q&A	Full Group	26/04/23	29/04/23	0%
5.3	Peer Assessment	Full Group			0%
5.4	Personal Statement	Full Group			0%

This Gantt chart includes a progress tracker for each task and a “Task Owner” that allows everyone to see who is doing what. This will be filled out as our project goes along and is updated regularly during our meetings. Some dependencies noted on the Gantt Chart: Aims & Objectives must be finished before the Development Summary is created, finish documentation before coding begins (Task 2 finished before Task 3 starts) and finish all tasks up to 4.4 before creating Presentation.

RISKS	CONTINGENCIES	LIKELIHOOD	IMPACT
Hardware failure	Regular backups of all data on Git to ensure no data is lost in case of hardware failure.	Low	High - can result in loss of data, delays in the project, and potential reputational damage if the hardware failure affects the project's users.
Software failure and programmin g problems	Regular backups of our program on Git. Allows the group to revert to previous versions of code if a software failure occurs. Regular code reviews to identify and resolve any programming issues.	High	High - it can result in bugs that affect the functionality of the project. This can lead to delays and increased costs.
Running out of time	Regular project status updates to ensure that the project stays on track. Regular reviews to identify and resolve any roadblocks or issues that could cause delays. Pair programming will also help with this as if one person is unable to complete their part of the project in time, their partner can step in.	High	Risk of running out of time can have a high impact, as it can result in missing the project deadline. This in turn will result in lower marks given even if the project is of good quality.
Security Issues	Regular security testing to identify and resolve any security risks in the program.	Medium	Can result in data breaches or other security incidents that can harm the users of our application.
Loss of backups	Regular backups of all data to multiple locations to ensure that backups are always available. We will be using Git to ensure our code is stored online and available at all times. We will also mainly use online document writers such as Google Docs to ensure that our documentation is safe as well.	Low	Can cause data loss, delays and increased costs if backups of the project are lost. This could in turn decrease the morale and productivity of the group.
Uncertainty in requirements	Regular requirements gathering and validation to ensure that the requirements are well-defined and understood by the group. Creating use-case diagrams and filling out the full requirements analysis will help with this.	Medium	Delays are the major thing that is an impact of the uncertainty of requirements. The quality of the project will also be lower which will result in a lower grade to be given.
Legal risks	Review the ethics and read about the legalities of our application to ensure that the project complies with all relevant laws and regulations.	Low	Can result in legal problems or penalties that can harm the project.
Integration Issues	Regular integration testing to identify and resolve any integration issues before they become major problems. Regular code reviews to identify and resolve any compatibility issues.	Medium	There could be delays and reduced user satisfaction because if the components do not work together seamlessly, the final product may not meet the user's expectations.

REFERENCES

[1] - Curry, D. (2021). *App Store Data (2021)*. [online] Business of Apps. Available at: https://www.businessofapps.com/data/app-stores/.

[2] - developer.apple.com. (n.d.). *Human Interface Guidelines - Human Interface Guidelines - Design - Apple Developer*. [online] Available at: https://developer.apple.com/design/human-interface-guidelines/guidelines/overview.

[3] - developer.apple.com. (n.d.). *Apple Developer Documentation*. [online] Available at: https://developer.apple.com/documentation/xctes