**Project Summary**

The project is a mobile application which provides a set of useful tools and features for people who use a narrowboat. It will be built on android and available for all android devices running android 10.0 and higher. The target audience for the app will be adults who use narrowboats, it is expected that a person using this app would have some knowledge and interest in narrowboats. Although the app is designed for narrowboats, it could also be used by any form boat.

One of the main features of the app is a logbook, almost all narrowboats have a form of logbook, either physical or electronic to keep a track on where they have been at what time. This app will provide them with a comprehensive logbook which will automatically keep track of where the user has been, the user will have full control over when the logbook is keeping track and when it is not. These entries within the logbook will form a boating session which is saved on the mobile device so it can be viewed later. These boating sessions can also be grouped together to form a journey which represents a collection of boating sessions. These sessions and journeys can also be searched and filtered allowing the user to see exactly where they have been and when, doing this for physical logbooks can be tedious depending on the size. Sessions and journeys will also have all the travelling data recorded such as a distance and time which can be useful to know.

Another feature of the app is to provide the user with an interactive map of the United Kingdom with all the canals shown on the map, taken from the open canals map [4]. This will allow users to see where they might want to go and allows for some route planning. The main feature of the map is allowing users to create and see points of interest on the map, these are canal locks and mooring locations, but the user can add their own such as a local water tap. These points of interest can have notes added to them, since the map is so large a user may find a point of interest has changed and may make a note of that on the map. The user can also choose not to share notes made on points of interest and use the system as a way of saving information about certain areas.

The app will also have a notification system which will alert the user of upcoming points of interest such as a dock is 1000 meters away. These notifications will popup on the user’s phone and will also be viewable on the app. The user can also opt into receiving notifications on canal stoppages and maintenance for a given area so if a canal is announced to be closed that you’re interested in they will be alerted straight away.

**Logbook Requirements**

* FRA1 – The user has the ability to start and stop a session. A session is when the user starts the logbook tracking and then stops it.
* FRA2 – The logbook must log their location at a given interval with timestamps during a session.
* FRA3 – The logbook will log at a 5-minute interval during a session.
  + FRA3.1 – The user can change this interval.
* FRA4 – The logbook must log what locks have been passed through with timestamps during a session.
* FRA5 – The user can filter the entries inside a session using a location.
  + FRA5.1 – The location filter will be a radius of any size in miles from a location of the users choosing. The location which the radius is centred on will be given as a name which google maps will find.
* FRA6 – The user can filter the entries inside a session using a period of time.
* FRA7 – The session will have a total amount of miles and hours travelled which can be seen by the user before the session has finished.
* FRA8 – The user can see past sessions which have been done.
  + FRA8.1 – Opening a previous session will show all the entries that were taken during that session.
* FRA9 – The user can filter sessions using a location.
  + FRA9.1 – The location filter will be a radius of any size in miles from a location of the users choosing. The location which the radius is centred on will be given as a name which google maps will find.
* FRA10 – The user can filter sessions using a period of time.
* FRA11 – The app will continue to record a session in the event that the connection to the internet is lost.
* FRA12 – The user can give a session a name, description, and change it later.
* FRA13 – The user can group sessions together forming a journey.
* FRA14 – The user can give a journey a name, description and change it later.

**Interactive Map Requirements**

* FRB1 – The user can see a map of the world with all the canals in the United Kingdom displayed.
* FRB2 – The map will have points of interests marked.
* FRB3 – The user can add notes to these locations of interest on the map.
* FRB4 – The user can add their own points of interest on the map if they are not already present.
  + FRB4.1 – A point of interest can be anything when the user is adding their own, it is not restricted to what is listed in FRC2.
* FRB5 – Other users of the app can see notes made about locations of interest.
* FRB6 – Other users can add notes to points of interest made by other users.
* FRB7 – Notes must be stored on the user’s phone and an online database if the user wishes other users to see their notes or other people’s notes / points of interest.

**Notification Requirements**

* FRC1 – The user will be able to choose what type of notifications they receive from the app.
  + FRC1.1 – Approaching a point of interest.
  + FRC1.2 – Canal maintenance and stoppages of multiple areas chosen by the user.
* FRC2 – The user will have a notification displayed on their phone when they approach a point of interest.
  + FRC2.1 – A lock is a point of interest.
  + FRC2.2 – A closed canal is a point of interest.
  + FRC2.3 – A mooring is a point of interest.
* FRC3 – The user will be able to see all these notifications on the app.

**Platform Requirements**

* FRD1 – The logbook must store all the session data on the phone using SQLite.
* FRD2 – The app must follow material design.
* FRD3 – The app must allow the user to navigate to other parts of the app with any type of android device.
* FRD4 – The app must support android devices running android 10+.
* FRD5 – The app must resize itself to support multiple resolutions and aspect ratios.
* FRD6 – The app will collect the canal data from the open canal map.
* FRD7 – The app will have a help page to teach users how the app functions.
* FRD8 – The app will contain links to useful websites regarding canals and narrow boats.
* FRD9 – The app will be compiled into an APK so it can be published and run-on android devices.

**Methodology**

To ensure this project is organised and follows a steady development lifecycle which is feasible a development methodology is best used. Since the project is to develop a piece of software for a mobile device, an agile methodology will be used. Agile development focuses on breaking the project into multiple development cycles which are relatively short when compared to the size of the project. This allows multiple iterations of the project to be developed with each iteration improving the product and incorporating more of the requirements. This also allows requirements to be added or changed later into the development life cycle without causing any issues.

The type of agile development which will be used is SCRUM. Since SCRUM is designed for a development team and not an individual, SCRUM for one will be used [1]. The aim of SCRUM is to encourage teamwork and organisation, it emphasises inclusivity and offering help to individuals who struggle with tasks. These ideologies can also be used by a single developer by self-reflection and product management. SCRUMS development lifecycle consists of sprints, these will be 1 weeklong and are the product iterations which agile development incorporates. At the start of a sprint, all the work which is to be completed for the sprint is picked out. Then a daily sprint is held where the current sprint is reviewed to see if the tasks are being finished and if the current sprint can be finished within the week. If a sprint is not feasible then tasks can be moved to the next sprint or be reviewed and changed. At the end of a sprint there is a sprint review to determine if it was successful. The constant self-reflection ensures that development is constant and steady for the product resulting in one which is constantly improving with no risk of not finishing it since requirements can be changed. It is also important to evaluate your mental health after each sprint to see if you can keep the current development lifecycle going, making notes on what went well and what went bad after each sprint will allow SCRUM to be adapted for my use.

To assist SCRUM, a Kanban bored, Trello [2], will be used. Incorporating a Kanban board into SCRUM allows progress for the project to be tracked easily. It also makes planning sprints easier as each requirement for the product can be broken down into many tasks and then these tasks can be used to build a sprint. If multiple tasks related to a requirement are not able to be completed, then that requirement which is causing issues can be identified and changed before it effects the development lifecycle.

When developing software, version control is important to ensure that progress is never lost, and the current version can be rolled back if new changes cause problems. As agile development requires weekly versions of the software to be produced having a version for each sprint is extremely useful. Github [3] will be used as version control for this project and is it free and easy to use.

**Researching other apps**

Nebo

* Tutorial on startup
* User accounts
* Interactive Map
* Markers which can be added by users
* User feed so you can share what you’ve been doing to others
* Message other users
* Live boat tracking of other users
* Prepare of voyage.
  + Exactly the same as what I want to do.
  + Has a log of where you’ve been with timestamps
  + Export voyage as pdf
* Good UI

RYA SafeTrx

* User accounts
* Weather reports in local area
* Information about the local area and warnings
* Safety guide built in
* Useful links page
* Prepare a journey
  + Exactly the same as what I want to do.
  + See a timeline which you can scroll like a video player to see where you were at what time.
  + Set tracking interval settings
  + Emergency contacts, call for help during journey.
* Can see past journeys
* Map doesn’t show anything other than your journeys
* Complicated UI when compared to Nebo but has a lot of useful tools
* Dark on dark colour scheme makes some stuff hard to see.

RCR Water Nav

* Don’t need account
* Have to download maps.
* Provides a map of the water canals.
* Map has markers for points of interest
* Very laggy

**Map Data**

The open canal map can be downloaded as a KMZ file which can be loaded by google maps to show all the canals and locks ect. Including this KMZ file within the android app should allow the app to have all the canals offline, whenever this map is updated a new KMZ can be downloaded when the app finds an internet connection, app shouldn’t need internet to function just GPS.

# Annotated bibliography

1. Raywenderlich (2017), Alex Andrews, “Scrum of One: How to Bring Scrum Into Your One-Person Operation” (Online) Available at: <https://www.raywenderlich.com/585-scrum-of-one-how-to-bring-scrum-into-your-one-person-operation> Accessed: 12th June 2022.

*An article detailing how SCRUM can be adapted for one-person development teams.*

1. Trello (2022), “Online Kanban board” (Online) Available at: <https://trello.com/> Accessed: 12th June 2022.

*An online and free Kanban board.*

1. Github (2022), “Version control and file hosting service” (Online) Available at: <https://github.com/> Accessed: 12th June 2022.

*Provides an extensive suite of tools for version control and file hosting with backups.*

1. Open Canals Map (2022), “Map of the United Kingdom canals” (Online) Available at: <https://opencanalmap.uk/> Accessed: 12th June 2022.

*Provides a complete map of the canals in the United Kingdom using google maps.*