

Statement of work

Team Details

Name: Possible Team

Team Leader: Jamie Westerhout

Name	Student No.	Degree	Skills/Experience/Interests
Marilyn Shao	43442148	Master of Information Technology	Web dev/design, software development, database
Marco Libera	47421815	B Computer Science	Web dev/databases/low level programming
Ella Barthelot	46420781	Bachelor of Information Technology / Science (Psychology)	Web/Mobile Dev (React), Design
Elie Gomah	47041017	B Computer Science(Cybersecurity)	Web design, app design/ web dev/ Penetration testing
Randy Abou Chakra	47015746	B Science (Physics) / B Computer Science (Machine Learning)	Data analysis/ machine learning/ data visualization
Jamie Westerhout	47451933	B Computer Science (Machine Learning)	Machine learning, software development, data visualization, data analytics

Project Outline

The project is to develop a mobile application to provide users with real time updates about flood conditions in the Brisbane area. It will feature a user interface displaying maps where users can view their location on the map, hazards in the area, and the locations of their family members or selected people. Hazards are reported by users by selecting a location, uploading an image of the hazard, and adding a title. The map will also display risk zones in which a high number of hazard reports have been made, where the number of reports indicate the likelihood of danger. This information will be used in conjunction with information gathered from official sources such as Brisbane City Council, QLD Search and Emergency Services, or QLD Police, to provide users with location specific information about flood conditions in their area

Features

Core (MVP) Features

Map View	User Reporting	Notifications
<ol style="list-style-type: none">1. User's can see their own location on a map, as well as the location of their family members.2. User's can see currently flooded areas (using data from 3rd party APIs e.g. from the BOM)3. Map is live and constantly refreshing with up to date data (fast)4. Check in feature for users to send quick notifications to family members in flooding areas that resets every hour.	<ol style="list-style-type: none">1. Users can mark specific areas (e.g. roads, buildings, parks etc) as flooded.2. Users can upload images to the app to show the conditions in specific areas.3. Information verification of user reports (using information from multiple user to increase confidence of reports with a combination of data from official s	<ol style="list-style-type: none">1. User's are notified if the BOM issues a flood warning for an area that they are either currently in or have marked as important.2. If there is a large volume of user reports of a hazard in there area
Optional/Extended Features	Potential Third Party API's	
<ul style="list-style-type: none">- Predicted flooding information using ml model- Offline availability- Customisation options - customization of alerts	<ul style="list-style-type: none">- X API from official sources such as SES, QLD Police, QFES- BOM RSS feed- Flood awareness map datasets from Brisbane- City Council	

Technical Requirements

Backend Tools	Frontend Tools
Python Flask for api hosting (session management) (Rest API) Gunicorn for WSGI server Nginx for web server Using requests python library for external api requests Fernet for encryption (password, session keys) SSL/TLS for encryption of information between client and server	React Native Expo on top of react native Gets all information from Our API
Backend Features/technical details	Frontend Features/technical details
Create Rest API that is accessible from the mobile that can deliver all the required data to the app Setup external api fulling then have that interface through a python class Setup relational database and have also accessible through a python class Setup server (flask) that handles logins, encryption, session... etc as well as delivering information to the app and then receiving calls from the app which tie into the external api class and database class	Google Maps API to display map React-native maps for integrating Google Maps into app Geolocation API to display the user's current position on the map React Navigation manages user navigation through the app React Context/Redux ensures real-time updates of family members' locations are reflected in the UI React Native Push Notifications React Native Elements to import icons

Project Plan

Timeline and personnel allocations

Front End			
Design Phase			
Week	Goal 1 – Marilyn	Goal 2 - Ella	Goal 3 - Elie
2	Desktop Research. Define Features of the App. Low Fidelity Wireframes.	Desktop Research. Define Features of the App. Define Aesthetics.	Define Features of the App. Desktop Research. Wireframe User Survey.
3	High Fidelity Prototype - Flood Map & Report screens. Initial Technical Research. Choose final tools/libraries	High Fidelity Prototype - Family connection screens. Setup environment.	High Fidelity prototype - Login & Setting screens. App logo design. Prototype User Survey.
4	Refine High Fidelity Prototype - Flood Map & Report screens based on user feedback.	Refine High Fidelity Prototype - Family connection screens based on user feedback.	Refine High Fidelity Prototype - Login & Setting screens based on user feedback.
5	Finalize High Fidelity Prototype.	Finalize High Fidelity Prototype.	Presentation preparation (with help from all members)
6	Mid Project Presentation & Prototype		
Development Phase			
7	Initial development of Flood Map & Report screens. Integrate Google Maps API and backend data.	Initial development of Family Connection screens. Integrate backend for real-time data updates of family member locations.	Initial Development of Login & Setting screens. Implement user authentication and basic settings functionalities.
8	Continue development of Flood Map & Report screens. Implement UI/UX refinements.	Continue development of Family Connection screens. Implement UI/UX refinements.	Continue Development of Login & Setting screens. Implement UI/UX refinements.
9	Conduct end-to-end testing for Flood Map & Report screens. Optimize performance and UI/UX based on feedback.	Conduct end-to-end testing for Family Connection screens. Optimize performance and UI/UX based on feedback.	Gather user feedback. Finalize Development of Login & Setting screens. Conduct comprehensive testing and refine based on feedback.
10	Final adjustments for Flood Map & Report screens. Prepare for final	Final adjustments for Family Connection screens. Prepare for	Final adjustments for Login & Setting screens. Prepare for final

	testing and deployment. Implement optional features.	final testing and deployment. Implement optional features.	testing and deployment. Implement optional features.
11	Conduct final testing. Fix any remaining bugs and prepare for expo.		
12	Final preparations for expo. Ensure the application is polished and fully functional.		

Back End (pre mid week -> working on information availability)			
Week	Goal 1 (Marco)	Goal 2 (Randy)	Goal 3 (Jamie)
2	Research and deciding on basic framework on how backend is going to be setup		
3	Set up initial database	initial setup to pull data from external api	Getting data accessible through the api (send and receive data), start working how to get logins to work and session ids
4	Database accessible from python to interrogate with goal 3, Nginx and gunicorn setup.	API pulling data from external sources and start interface to make them accessible to the server, research for more sources	Finish login system and session tokens working, setup base for notification requests, and external api request (define structure for them)
5	Web Server fully setup for flask (api server) to hook into, and database schema developed and fully accessible through python	Python interface setup so it can be info incorporated into our api and working, and continue research and adding more external sources, get region resolution with map api working so coordinates can be associated with region	Get database information accessible through the api and make it so necessary information is stored. As well as setting up pipeline between external api interface and the api storing some information in database
6	Mid Project Presentation & Prototype (post mid pres -> working on data verification)		
7	Refine database performance and interactions through python,	Help with trust metric and data verification -> ensuring official sources information is recent and any outdated data is discarded	Setup stacking of hazard reports and giving metric of how trustable the report is using the amount of similar user reports made
8	Setup system to figure out when hazard reports are a certain age and gives list of reports about to expire (in addition to setting up report expirations)	Set up system to try and match official reports to user reports that will add a correlation between user and official source reports	Set up system to send reports that are about to expire to user close to the reported area and the original reporter to check to see if the report is up to data
9	Finishing of aging and making sure database size and information pull times are good	Finish of external source information extraction and external and user report matching	Finish of trust metric and user report matching
10	Assuming everything goes to plan -> week dedicated to implementing optional/extended features		
11	Polishing off everything, bug fixes make sure code is clean, full commented up to standard		
12	Prepare for expo		

Milestones and Deliverables

Milestones	Deliverables
<ul style="list-style-type: none"> - Implement database based on requirements analysis and schema developed - Database accessible through API - Integration of external data sources into back end - Connect backend API with mobile app 	<ul style="list-style-type: none"> - Statement of work outlining project arrangements - Functional database based on defined schema - Functional backend supports all required features - Functional mobile application fully integrated with backend with appropriate UI/UX applied

Assumption

Technology availability & Third-party API reliability	Privacy legal requirements when required
Team availability	User feedback in timely manner
All family members/friends will be users of on the app for location tracking and check ins	Sufficiently large user base

Risk Matrix

Possible Risk	Type	Likelihood	Severity of Impact	Minimisation strategy
Personal Information Leak	Data privacy	Medium	High	Only require minimal information for sign up. Other information like “location” will not be kept on servers. Location just flows through the server and is stored on devices.
External actors wanting to harm the app by spamming the “user data” with false information (this can be done at large using bots)	Security	Low	Medium	Require identification such as an email or a phone number.
Team member dropping out for any particular reason	Project	Low	Medium	Group members have been chosen in such a way that our skill sets are different enough so that we can all contribute, but also with enough similarity such that if a team member drops out, others can help take over. In addition, all group members are comfortable with adapting to new skills and learning things outside of their main skill sets.
Basing flood risk results on wrong or outdated information, thus potentially causing unnecessary stress for app users and possibly endangering them.	Ethical	Low	High	Utilize data from well known and reliable external sources. In addition, using data from the users themselves (user reports)

Collaboration Plan

Collaboration tools	Meeting plan
<ul style="list-style-type: none">- Microsoft Teams- Github projects for project management- Github for source version control/code collaboration	<ul style="list-style-type: none">- Every friday in studio- Online teams meeting throughout the week as needed (wednesday night everyone is normally available if an extra meeting is required)