

Software Development Plan

4.1 Plan Introduction.....	1
4.1.1 Project Deliverables.....	2
4.2 Project Resources.....	4
4.2.1 Hardware Resources.....	4
4.2.1.1 Development Machine.....	4
4.2.2 Software Resources.....	4
4.3 Project Organization.....	5
4.3.1 Frontend engineer(s).....	5
4.3.1.1 UI Designer.....	5
4.3.1.2 UI Engineer.....	5
4.3.1.3 Frontend Engineer.....	5
4.3.2 Backend engineer.....	5
4.3.2.1 Database Designer.....	5
4.3.2.2 Database Engineer.....	5
4.3.2.3 Server Architect.....	5
4.3.2.3 Server Engineer.....	5
4.4 Project Schedule.....	6
4.4.1 PERT / GANTT Chart.....	6
4.4.2 Task / Resource Table.....	6

4.1 Plan Introduction

This software development plan outlines the details for the development of Visor, an application that allows users to find items they are looking for by asking for help.

Visor will allow users to post questions concerning an item(s) they seek and other users will respond with instructions on how to find the item(s). The aim of Visor is to solve the problem of people wandering about in places looking for specific items whose whereabouts they do not know. With Visor, users can leverage their community to get help and find things inside places (buildings) quicker.

Visor's development will include code planning, implementation, testing, documentation, and UI design.

Project Milestones

Milestone	Due Date
Open Camera Button	2024-01-12
Bug Fix: Checking into Places	2024-01-19
Bug Fix: Load time for async images too long	2024-01-26
Bug Fix: Reloading Posts every time view appears	2024-01-31
SimilarQuestions View	2024-02-19
PlacesWithItemView	2024-02-29
Replying to replies	2024-03-15
Upvote/Downvote count	2024-03-22
BugFix: Date posted formatting	2024-03-31
Persistence	2024-04-30

4.1.1 Project Deliverables

This section shows the project deliverables with more detailed explanations.

- ❖ Open Camera Button

- Allow the user to click a button that opens the user's camera so they can take a picture of an unknown item. The button click should request user permission to use their camera if the user has not yet given such permissions to the application. This functionality will allow the user to take pictures of unknown items they are looking for so that Visor can identify them using Google Lens' API.

- **Due date: 2024-01-12**

- ❖ Bug Fix: Checking into Places

- Application constantly asks the user to check into a place even after the user has declined to do so or already checked in. The aim of this bug fix is to prevent the application from annoying users with constant check in requests. The application should request the user to check in once after launch and only ask again if the user moves to a different place.

- **Due date: 2024-01-19**

- ❖ Bug Fix: Load time for async images too long

- Remote images take too long to load. This bug fix aims to reduce the load time for remote images so they can be displayed quicker.

- **Due date: 2024-01-26**

- ❖ Bug Fix: Reloading Posts every time view appears

- Posts constantly reload when a view reappears, making unnecessary requests to the server. This bug fix aims to stop these unnecessary requests and only load posts that have not already been loaded when users scrolls to the bottom of the screen.

- **Due date: 2024-01-31**

- ❖ SimilarQuestions View

- This view will allow users to see similar questions to the one they asked. This way, the user does not have to wait for other users to respond to their question to get answers. Instead, they can see similar questions asked and hopefully find the answer they seek.

- **Due date: 2024-02-19**

- ❖ PlacesWithItemView

- This view will allow users to see places that have the item they seek.

Implementation of this will require external API calls to Google Search. The aim is to allow users to see places they can find items they seek without waiting for responses from other users.

- **Due date: 2024-02-29**

- ❖ Replying to replies

- Functionality will enable users to respond to a response and see responses to a response. The aim is to allow users to continue conversations in the Comments section and agree on a solution (most appropriate answer).

- **Due date: 2024-03-15**

- ❖ Upvote/Downvote count

- Show the net of upvotes minus downvotes on a response/post. This will be an indicator to other users on whether they should engage with the post/response. Negative values will mean the general consensus is that this is not an appropriate post/response.

- **Due date: 2024-03-22**

- ❖ BugFix: Date posted formatting

- Dates on posts currently show as the exact date a post was made (example Feb 2nd, 2023). The aim of this bug fix is to change this to relative dates to make it easier for users to glimpse how long ago a post was made. An example of the desired date format is “2 months ago.”

- **Due date: 2024-03-31**

- ❖ Persistence

- The application currently does not have data persistence. It asks users to login every time the app is launched and reloads posts and data from the server. Adding persistence to the app will entail storing necessary information in the user's device. Data that will be persisted will include: user login sessions, a sufficiently small number of posts that were loaded on a previous launch and any other data that does not constantly require remote server calls on every app launch.

- **Due date: 2024-04-30**

4.2 Project Resources

This section details the hardware and software resources required for completion of the project.

4.2.1 Hardware Resources

4.2.1.1 Development Machine

A Macbook Air with 1.1GHz processor and 16GB storage space will suffice for development. The Macbook might require macOS 14.3 or higher.

4.2.2 Software Resources

The following software versions will be used in development

- ❖ Xcode 15
- ❖ Swift 5 or higher
- ❖ SwiftUI 5
- ❖ Visual Studio Code 17.9 or higher
- ❖ NodeJs 21.6.2 or higher
- ❖ ExpressJs 4.18.2
- ❖ Node Package Manager (NPM) 10.4.0
- ❖ React Native 0.73.4
- ❖ SerpAPI
- ❖ Google Places API
- ❖ Firebase (10.21.0 iOS SDK), (10.8.0 Javascript SDK)
- ❖ JavaScript (ES13)

4.3 Project Organization

4.3.1 Frontend engineer(s)

The frontend engineers will be responsible for designing and implementing the user interface.

4.3.1.1 UI Designer

The UI designer will design the User Interface (UI) elements of the application. UI elements include any component, buttons, pictures, texts, etc, that will be rendered on the screen.

4.3.1.2 UI Engineer

The UI engineer will implement the elements designed by the UI designer in code.

4.3.1.3 Frontend Engineer

Frontend engineer will be responsible for implementing functions in the client side of the app that are not visible to the user.

4.3.2 Backend engineer

The backend engineer(s) will be responsible for designing and implementing the backend of the application (database and server).

4.3.2.1 Database Designer

The database designer will be responsible for designing the database. This will include detailing the collections in the database, what types of documents each collection will store and how the collections will be linked to each other.

4.3.2.2 Database Engineer

This person will be in charge of writing code that performs CRUD operations on the database.

4.3.2.3 Server Architect

This person will design the server, detailing the API routes that will be used for specific calls to the servers and which controllers will respond to which API call and the fields in the JSON that will be sent as a response, if any.

4.3.2.3 Server Engineer

This person will implement in code the routes and controllers designed by the server architect.

4.4 Project Schedule

4.4.1 PERT / GANTT Chart

The Gantt chart can be found at [✚ Visor Gantt Chart](#)

4.4.2 Task / Resource Table

Task	Resource(s) Needed	Human Resource
Open Camera Button	Macbook, Swift, Swift UI, Xcode	UI Designer, UI Engineer
Bug Fix: Checking into Places	Macbook, Xcode, Swift	UI Engineer
Bug Fix: Load time for async images too	Macbook, Xcode, Swift	UI Engineer

long		
Bug Fix: Reloading Posts every time view appears	Macbook, Xcode, Swift	UI Engineer
SimilarQuestions View	Macbook, Swift, Swift UI, Xcode	UI Designer, UI Engineer
PlacesWithItemView	Macbook, Swift, Swift UI, Xcode	UI Designer, UI Engineer
Replying to replies	Macbook, Swift, Swift UI, Xcode, ExpressJs, Visual Studio, JavaScript, Firebase	UI Designer, UI Engineer, Database Designer, Database Engineer, Server Architect, Server Engineer
Upvote/Downvote count	Macbook, Swift, Swift UI	UI Designer, UI Engineer
BugFix: Date posted formatting	Macbook, Swift, Swift UI	UI Designer, UI Engineer
Persistence	Macbook, Swift, Xcode	Frontend Engineer