Requirements Specification Document Visor

Table of Contents

1 Scope

1.1 Identification

This Software Requirements Specification (SRS) documents the requirements for the application Visor.

1.2 System Overview

The Visor Application is made to enable people to find whatever they are looking for, wherever it is. It implements a question and answer service, much like StackOverflow, that allows users to ask questions about items they need and want and get responses from knowledgeable users.

Visor is made with both Swift and React Native, and it is available on both iOS and Android. The application has various views that enable users to add questions, view responses to questions, and a settings view to make changes to their preferences.

1.3 Document Overview

This document is organized as follows. Section 2 contains an in-depth description of the system operation. Section 3 provides details of computing, compiling, development, and other supporting resources required. Section 4 explains the qualification methods that will be used and how each requirement will be tested with these methods as well as the distribution provisions necessary to distribute the software. Section 5 is the appendix section and defines the acronyms used throughout the document. Section 6 lists related documents used across Visor that may be of interest to the reader.

Table 1- Document Definitions

Type	Definition

Shall	Expresses a mandatory provision.
Should	Expresses non-mandatory provision.
Will	Declaration of purpose such as a design goal.

2 System Operation and Requirements

2.1 System Divisions

Visor is divided into three separate components: frontend, server (backend), and database. The frontend is responsible for rendering the application to the end user and allowing user interaction. The server shall handle all network requests necessary to make the application function. The server is the bridge between the frontend and the database. The database shall house all necessary information needed to make Visor function.

2.2 Frontend

2.2.1 The frontend shall be made with Swift UI (for iOS) and React Native (Android).
2.2.2 The frontend shall consist of five interconnected views
2.2.3 The first view shall allow the user to sign into the application
This view will be known as the SignInView
2.2.4 The SignInView shall make network requests to the server to authenticate the user
2.2.5 A second view shall allow the user to view all questions asked on Visor
This view will be known as the HomeView.
The HomeView should show posts closest to the user's location first.

The HomeView should order posts by date posted, with recent posts taking precedence over older posts.

2.2.6 A third view shall allow users to view responses to questions

This view will be known as the CommentsView.

2.2.7 A fourth view shall allow the user add a new question (post)

This view will be known as the AddPostView

2.2.8 A fifth view shall allow the user to edit their settings and preferences

This view will be known as the SettingsView.

The Settings view should allow the user to log out.

The Settings view should allow the user to delete their account.

2.3 Database

2.3.1 The Database shall store necessary information vital for the functionality of Visor

2.3.2 The Database shall be securely implemented to prevent malicious attacks

2.3.3 The Database should store both text based information and media (videos and pictures)

2.4 Sever

2.4.1 The server shall handle all network requests to interact with information in the database and send it to the user

The server should be separated into Controllers and Routes

Routes should define the specific API endpoint and type of HTML request (GET, PUT, DELETE, PATCH etc) that the server must respond to

The controllers should implement the functions that respond to a specific server route

The controllers should return the appropriate HTML response status code and an easily interpretable JSON object to answer a request from the frontend

3 Computing Resource Requirements

3.1 Functional Requirements

3.1.1 Users shall be able to login

3.1.2 Users shall be able to view questions posted by other user

Questions will be sorted by distance from user (closest ones first) and date posted (most recent posts first)

3.1.3 Users shall be able to post a new question

3.1.4 Users shall be able to upvote and downvote a question

3.1.5 Users shall be able to view responses to a question

3.1.6 Users shall be able to resolve a question

3.1.7 Users shall be able to unresolve a question

3.1.8 Users shall be able to delete a question they posted

3.1.9 Users shall be able to create a new account (if they do not have one)

3.2 External Interface Requirements

No additional requirements.

3.3 Software Item Internal Interface Requirements

No additional requirements.

3.4 Software Item Internal Data Requirements

No additional requirements.

3.5 Adaptation Requirements

No additional requirements.

3.6 Safety Requirements

No additional requirements.

3.7 Security and Privacy Requirements

No additional requirements.

3.8 Operating Environment

3.8.1 Computing Hardware Requirements

3.8.1.1 Visor shall require an iPhone (iOS version) and/or an Android (Android version) to work. Emulators can be used in lieu of physical devices for testing.

3.8.2 Computing Software Requirements

3.8.2.1 iOS version of Visor shall be able execute in a standard iOS environment

3.8.2.2 Android version of Visor shall be able execute in a standard Android environment

3.9 Software Quality Factors

No additional requirements.

3.10 Design and Implementation Constraints

No additional requirements.

3.11 Personnel-related Requirements

No additional requirements.

3.12 Training Requirements

No additional requirements.

3.13 Logistical Requirements

No additional requirements.

3.14 Other Requirements

No additional requirements.

3.15 Packaging Requirements

No additional requirements.

3.16 Precedence and Criticality of Requirements

No additional requirements.

4 Qualification Provisions

4.1 Qualification methods

Requirements shall be qualified by analysis (A), inspection (I), test (T), demonstration (D) or special (S) as described in the table below.

Qualification Code	Qualification Method	Description
A	Analysis	The processing of accumulated data obtained from other qualification methods. Examples are reduction, interpretation, or extrapolation of test results.
I	Inspection	The visual examination of software item code, documentation, etc.
T	Test	The operation of the software item, or a part of the software item, using instrumentation or other special test equipment to collect data for later analysis.

D	Demonstration	The operation of the software item, or a part of the software item, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
S	Special	Any special qualification methods for the software item, such as special tools, techniques, procedures, facilities, and acceptance limits

4.2 Qualification matrix

The table below contains each requirement in section 2.0 and the method of verification.

Paragraph	Requirement	Compliance Synopsis	Verif. method
2.2.1	The frontend shall be made with Swift UI (for iOS) and React Native (Android)	Observe that the code is written in Swift and React Native.	I
2.2.2	The frontend shall consist of five interconnected views	Observe that the application has 5 views.	I
2.2.3	The first view shall allow the user to sign into the application.	Demonstrate that the user can login with their credentials.	D
2.2.4	The SignInView shall make network requests to the server to authenticate the user	Demonstrate that the application can make network requests to authenticate the user.	D
2.2.5	A second view shall allow the user to view all questions asked on Visor	Observe that the HomeView shows all questions asked on visor.	I
2.2.6	A third view shall allow users to view responses to questions	Demonstrate that a user can view responses to questions	D

2.2.7	A fourth view shall allow the user add a new question (post)	Demonstrate that a user can post a new question	D
2.2.8	A fifth view shall allow the user to edit their settings and preferences	Demonstrate that a user can access settings and change their preferences	D
2.3.1	The Database shall store necessary information vital for the functionality of Visor	Observe that the database is able to hold information	I
2.3.2	The Database shall be securely implemented to prevent malicious attacks	Demonstrate that the database is able to guard against malicious attacks	D
2.3.3	The Database should store both text based information and media (videos and pictures)	Demonstrate that the database can store text bases information, pictures, and videos	D
2.4.1	The server shall handle all network requests to interact with information in the database and send it to the user	Demonstrate that the server can perform CRUD operations on the database and respond to user requests	D

4.3 Distribution Provisions

Visor will be distributed as an application available for download on both the App Store and Google Play Store. The server will be hosted remotely.

Users only need to download and install the application to be able to use it. The application can be used on all iOS and Android devices that are compatible with the minimum iOS and Android target SDKs. The minimum iOS and Android target SDKs are yet to be decided.

5 Appendices

5.1 Acronyms

Acronym	Definition/Description
CRUD	Create, Read, Update, Delete
HTML	Hyper Text Markup Language
iOS	iPhone Operating System
JSON	JavaScript Object Notation
UI	User Interface
SDK	Software Development Kit

6 Related Documents

- Bug Fix: Checking into places
- Seeing other Places Design Doc