Software Requirements Specification

BeenThereDoneThat

Christina Choi
Troy Womack-Henderson

February, 7th, 2024

Table of Contents

Table of Contents.....2

5.1 Introduction..... 3

Fig 5.1.1 System Component UML Diagram..... 3

5.2 Functional Requirements..... 4

5.2.1 View - Graphical User Interface.....4

5.2.2 Model - Database.....4

5.2.3 Controller - Brains..... 4

5.3 Performance Requirements..... 5

5.3.1 User Authentication..... 5

5.3.2 Journal Entry Submission..... 5

5.3.3 Recommended Prompts Database..... 5

5.4 Environment Requirements..... 5

5.1 Introduction

BeenThereDoneThat is a mobile application that is a journaling platform aimed at providing users with a convenient way to capture and reflect on their daily experiences. The system facilitates the creation, management, and exploration of journal entries accompanied by photos, prompts, and other optional features. The app operates on Apple's iOS operating system and requires internet connectivity for synchronization.

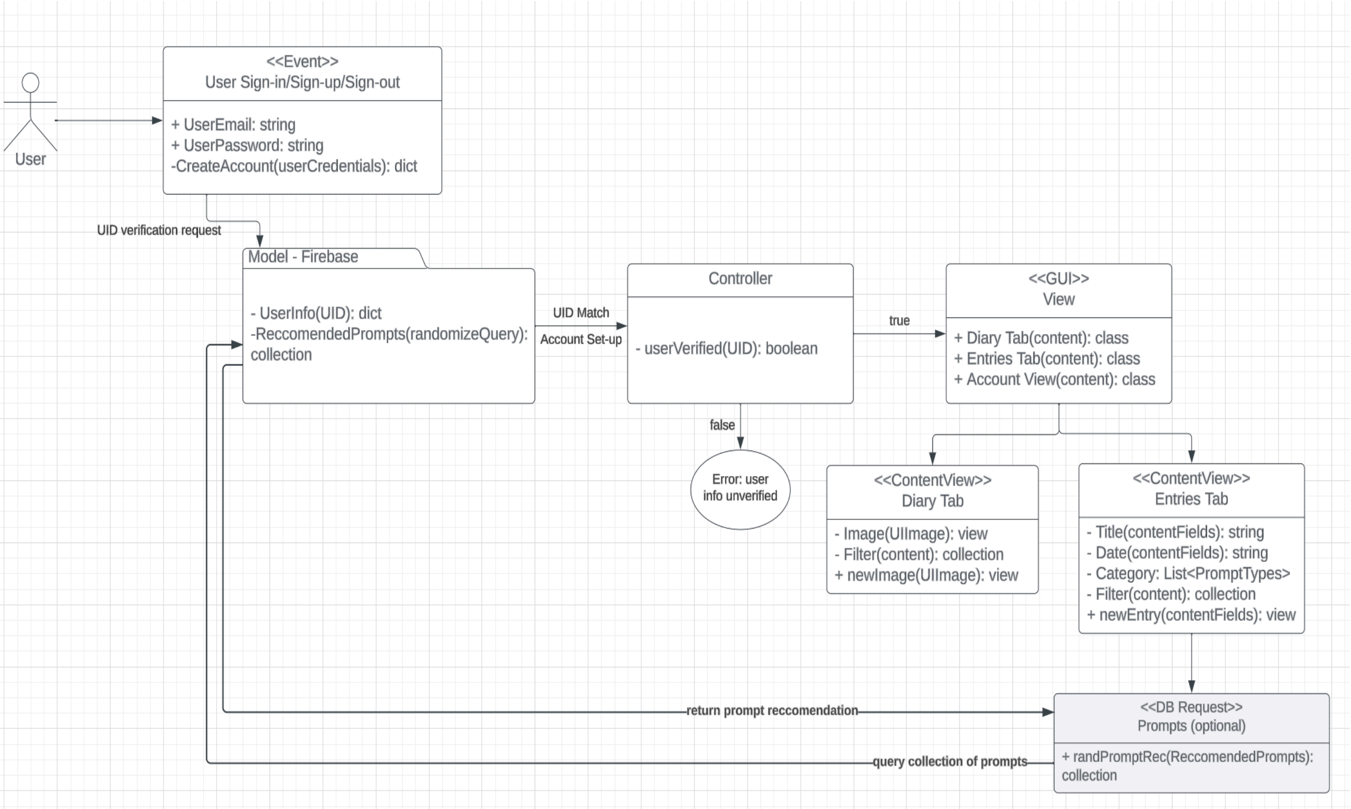
User
<<Event>>
User Sign-in/Sign-up/Sign-out
+ UserEmail: string
+ UserPassword: string
-CreateAccount(userCredentials): dict
UID verification request
Model - Firebase
- UserInfo(UID): dict
-ReccomendedPrompts(randomizeQuery): collection
UID Match
Account Set-up
Controller
- userVerified(UID): boolean
true
false
Error: user info unverified
<<GUI>>
View
+ Diary Tab(content): class
+ Entries Tab(content): class
+ Account View(content): class
<<ContentView>>
Diary Tab
- Image(UIImage): view
- Filter(content): collection
+ newImage(UIImage): view
<<ContentView>>
Entries Tab
- Title(contentFields): string
- Date(contentFields): string
- Category: List<PromptTypes>
- Filter(content): collection
+ newEntry(contentFields): view
return prompt reccomendation
query collection of prompts
<<DB Request>>
Prompts (optional)
+ randPromptRec(ReccomendedPrompts): collection

Fig 5.1.1 System Component UML Diagram

In this Software Requirements Specification Document, the following document is structured as follows. Section 5.2 contains the functional requirements, as in the features that are expected to be in this mobile app, to develop *BeenThereDoneThat*. Section 5.3 contains the performance requirements, as in requirements we expect the application to accomplish in the app. Finally, Section 5.4 contains the environment requirements, which are the software and hardware needed to deploy this app on the user's end.

5.2 Functional Requirements

The functional requirements section outlines the specific features and functionalities that the *BeenThereDoneThat* app will provide to its users. These requirements detail the behavior and capabilities of the app's graphical user interface (GUI) in facilitating the creation, management, and access of journal entries. By clearly defining these functional requirements, the app should meet the users' expectations for usability, accessibility, and overall user experience.

5.2.1 View - Graphical User Interface

In BeenThereDoneThat, the graphical user interface will provide an easy and concise way to upload pictures and journal entries.

5.2.1.1 The GUI shall provide two tabs that the user can flick through for the two types of journal entries - Picture Diary and Journal Entry. The tabs shall be on the bottom of the screen with an icon to easily differentiate between the two types.

5.2.1.2 The GUI shall provide various button widgets for the user to create and maintain their entries. There shall be a filter button for the user to filter through all the entries they have created by date (oldest, newest) and title of the entry (A-Z and Z-A). There shall be a plus button in an easily accessible location for the user to upload picture diaries and create a new journal entry.

5.2.1.3. The GUI shall provide the user access to their account once they receive a green light from the controller. If the email is not in the database or the password is wrong, it shall pass an error.

5.2.2 Model - Database

Our application will utilize FireBase, storing user authentication (email and password), the history of the user’s submitted journal entries, and a collection of all recommended prompts for the users to flick through and choose the one of their preference.

5.2.2.1 When the controller requests FireBase for user authentication, FireBase shall run that request, looking for that email and seeing if the password associated with that email matches the one the user inputted. If it does, it passes the data along to the controller.

5.2.2.2 When the controller requests FireBase to filter past submitted journal entries, FireBase shall run that request, filter the database of entries accordingly, and pass the data to the controller.

5.2.3 Controller - Brains

The controller serves as the intermediary between the mobile application and FireBase, facilitating data exchange and application logic.

5.2.3.1 The controller shall connect the mobile application and FireBase when a user logs in or signs up. When the user clicks login and enters their email and password, the controller shall convert that event into a request to FireBase to look through their user authentication database. When the controller receives the data from FireBase, it shall pass that data onto the GUI.

5.2.3.2 The controller shall connect the mobile application and FireBase when a user chooses one of the filtering options. The controller shall convert this event into a request to FireBase.

5.3 Performance Requirements

The Performance Requirements section will outline the expectations for the performance of all the core functionalities of the application and all its possible outcomes use cases.

5.3.1 User Authentication

5.3.1.1 For a user's first time signing up, the application shall create a new identifier row in the user authentication database in FireBase. It will include the user's email, the date they created their profile, the date they last signed in, and a unique user UID to ensure they don't make multiple accounts. After the sign-up, whenever the user tries to log in to their account, FireBase ensures they enter the right password to access their account.

5.3.2 Journal Entry Submission

5.3.2.1 The application shall create a new "document" in the Firestore Database under the "Journal Entries" collection. This document will contain the fields: "category" (which category this journal entry belongs to), "prompt" (title of the entry), and "date" (the date this entry was written).

5.3.3 Recommended Prompts Database

5.3.3.1 The "Recommended Prompts" collection in Firestore shall return a list of recommended prompts, sorted into groups by their categories. The only field in the documents shall be their category.

5.4 Environment Requirements

Hardware Requirements for *BeenThereDoneThat*:

Category	Requirement
Processor	Intel i5- or i7-equivalent CPU

Hard Drive Space	40GB
RAM	8GB
Display	800x600, 256 colors

Software Requirements for *BeenThereDoneThat*:

Category	Requirement
Operating System	macOS
Compiler	Apple Clang Compiler
Graphics	XCode, Figma