Software Requirements Specification

BeenThereDoneThat

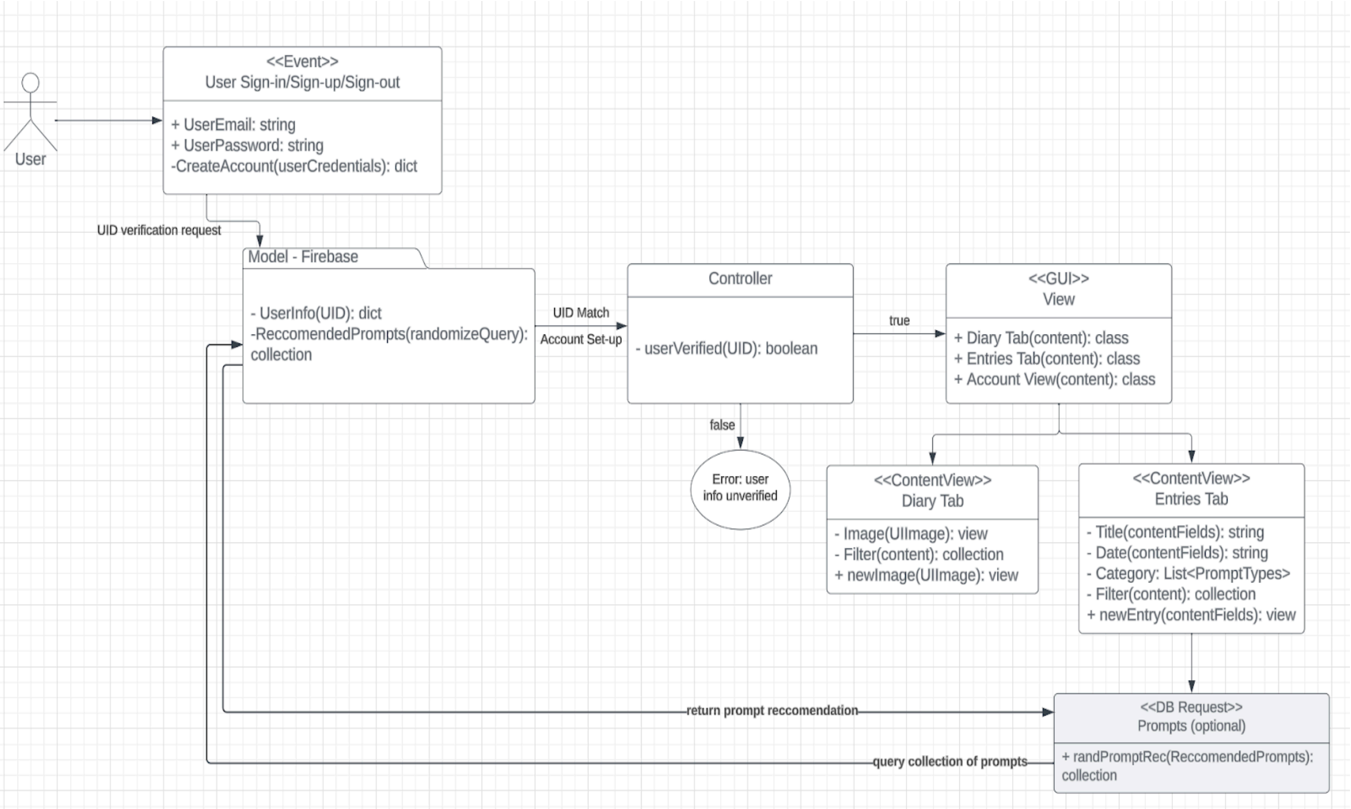
Christina Choi
Troy Womack-Henderson

April 3rd, 2024

Table of Contents

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
<<GUI>>
View
+ Diary Tab(content): class
+ Entries Tab(content): class
+ Account View(content): class
Error: user info unverified
<<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.
5.2.1.2 The tabs will have distinct icons positioned at the bottom of the screen to differentiate between picture diary and journal entry.
5.2.1.2 The GUI shall provide a filtering button widget for the user to organize their view of all entries in their journal.
5.2.1.3 The GUI shall provide a plus entry button widget for the user to upload new entries to their journal.
5.2.1.4. The GUI shall grant users access to their accounts, providing appropriate error messages for incorrect credentials.

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 The model CSU shall be implemented using Firebase.
5.2.2.2 The model shall run requests from the controller when prompted by the user.
5.2.2.2.1 One type of controller request shall be for user authentication.

5.2.2.3 The model shall run authentication requests by passing them to Firebase.

5.2.2.3.1 Authentication requests will include the user's e-mail address.

5.2.2.3.2 Authentication requests will include the user's password.

5.2.2.4 The model shall, using Firebase built-in functions, find records matching the e-mail address.

5.2.2.5 The model shall, upon matching a record to the e-mail contained in the authentication request, compare the password from the authentication request with the password from the matched record.

5.2.2.5 Properly matched authentication data shall be returned 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 facilitate communication between the mobile application and FireBase during login/signup processes.
5.2.3.2 The controller shall handle filtering options, converting user selections into requests to FireBase.
5.2.3.3 The application shall generate a unique identifier for each user during sign-up, ensuring one account per user and storing essential user details securely.
5.2.3.4 Journal entries shall be stored as documents in the Firestore Database, containing fields for category, prompt, and date.
5.2.3.5 The "Recommended Prompts" collection in Firestore shall provide categorized lists of prompts for user selection.
5.2.3.6 Prompts shall be grouped by category within the Firestore collection.
5.2.3.7 Performance requirements shall ensure seamless execution of all core functionalities and use cases.
5.2.3.8 User authentication processes shall be efficient and secure, allowing users to access their accounts promptly.

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 During user sign-up, the application shall swiftly create a new identifier row in the FireBase authentication database
5.3.1.2 The identifier shall store essential user details like the email, creation date, last sign-in date, and a unique user UID.
5.3.1.3 FireBase shall verify user credentials promptly during login attempts, ensuring only authorized users gain access to their accounts.

5.3.2 Journal Entry Submission

5.3.2.1 The application shall efficiently create new documents in the Firestore Database for journal entries.
5.3.2.2 The journal entry fields shall include a title and journal entry body with pre-saved time/date for record keeping of each entry’s attributes.

5.3.2.3 The journal entry posting process shall be optimized to minimize latency and ensure a seamless user experience.

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