Simon Fraser University
CMPT 275: Software Engineering I

Design Document

Version: 2
November 6, 2017

Group Number: 11
Group Name: Invincible

Group Members:
Jia Hui (Mandy) Xiao
Gary (Sheung Yau) Chung
Zavier Aguila
John Kyung Joon Ko
Ryan Kiew Ruelt Yean

Revision Chart

Version	Description of Version	Date Completed
1.0	Draft	Oct. 16, 2017
1.1	Added class diagram and data requirements	Oct. 18, 2017
1.2	Reviewed	Oct.19, 2017
2.0	Updated according to the development process	Nov. 2, 2017

Table of Contents

1 Introduction

This document is intended to outline the high-level organization of the system for the app "*Prep*". It includes all technical guidelines such as development frameworks and address any ethical or legal issues surrounding this app. Two system diagrams are provided. Namely the state diagram and the class diagram showing how the system work together. Furthermore, this document will also specify data requirements with descriptions of file formats and input/output of the system. Finally, feature priorities for each iteration are listed.

2 Guidelines

2.1 Operating Systems

macOS High Sierra 10.3, the latest Apple operating system, has all the required tools to create Prep.

iOS 11 is a mobile operating system used on the iPad, which is the platform Prep will be deployed on.

2.2 IDE

Prep will be built using XCode version 9.0

2.3 Frameworks

SceneKit is a graphics framework native to macOS which is used to generate 3D animated scenes. This framework also offers functionality for rendering panoramas, and placing 3D objects in a panorama, which will be the intended use for Prep.

SpriteKit is another framework similar to SceneKit, but deals with 2D animations. We will use SpriteKit for the implementation of the UI within the Experience Editor.

2.4 APIs

Cocoa is an API native to mac OS which will aid in creating the user interface for Prep.

2.5 External Libraries

OpenCV, version 2.4.13.4, is a library with many functions that is compatible with XCode. This library has many useful capabilities that will aid in creating Prep. It will be specifically used to stitch images a user uploads, which then will be rendered into an panorama environment.

2.6 Programming Languages

Swift, version 4, is the programming language our team has chosen to develop Prep.

C++, version C++11, will be used when dealing with OpenCV.

2.7 Legal and Ethical Issues

Prep will have a sound recording function for the purpose of associating audio to an object in a panorama picture. The issue that arises from this function is that if the app records a private conversation between people, that doesn't involve the user of the app, it's illegal to do so in the Canadian Criminal Code, R.S.C. 1985, c. C-46. Many other countries also follow the same adherence on sound recording like the United States. Moreover, it would be a breach of privacy and would cause a legal quandary. However, the legality of this can be circumvented in the app's terms of use document, which the user must comply with to use Prep. The terms of use should specify the user will not record private conversations that don't include themselves.

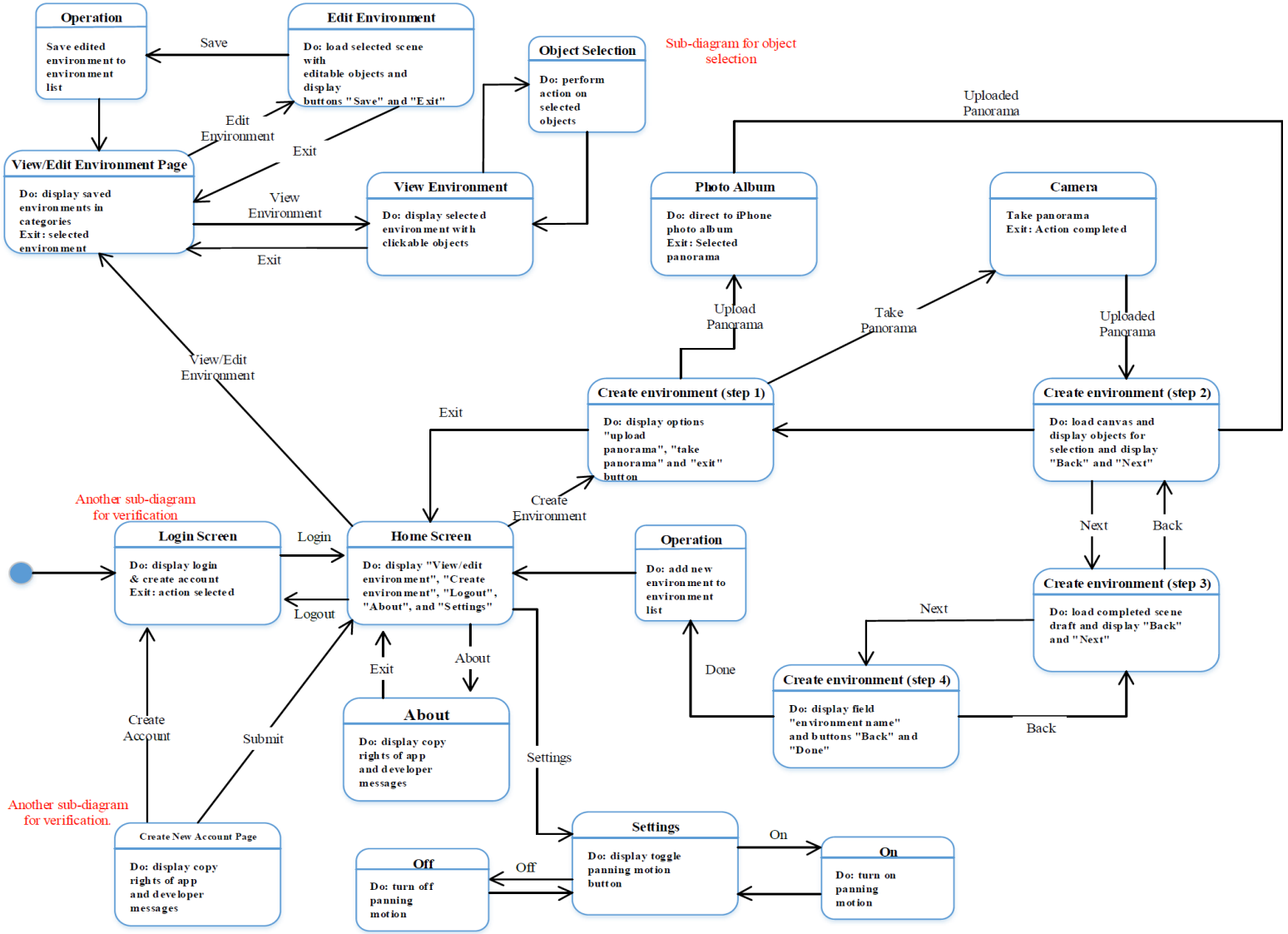
Users will be able to use pre-existing pictures, audio, and videos. Unfortunately, copyright laws exist, so do laws against uploading and possessing illicit materials. Users may upload copyrighted, or illegal material into the app which would pose grave consequences. However, again if we specify against copyrighted and illegal materials in the terms of use our team would have no obligation of negligent use.

Although our team believes Prep is a helpful tool, to ease sensory overload, and anxiety from being in new surroundings, it should not replace a healthcare professional's advice and recommendations. It's in the user's best interest to use Prep alongside the activities recommended by a physician.

3 System Diagrams

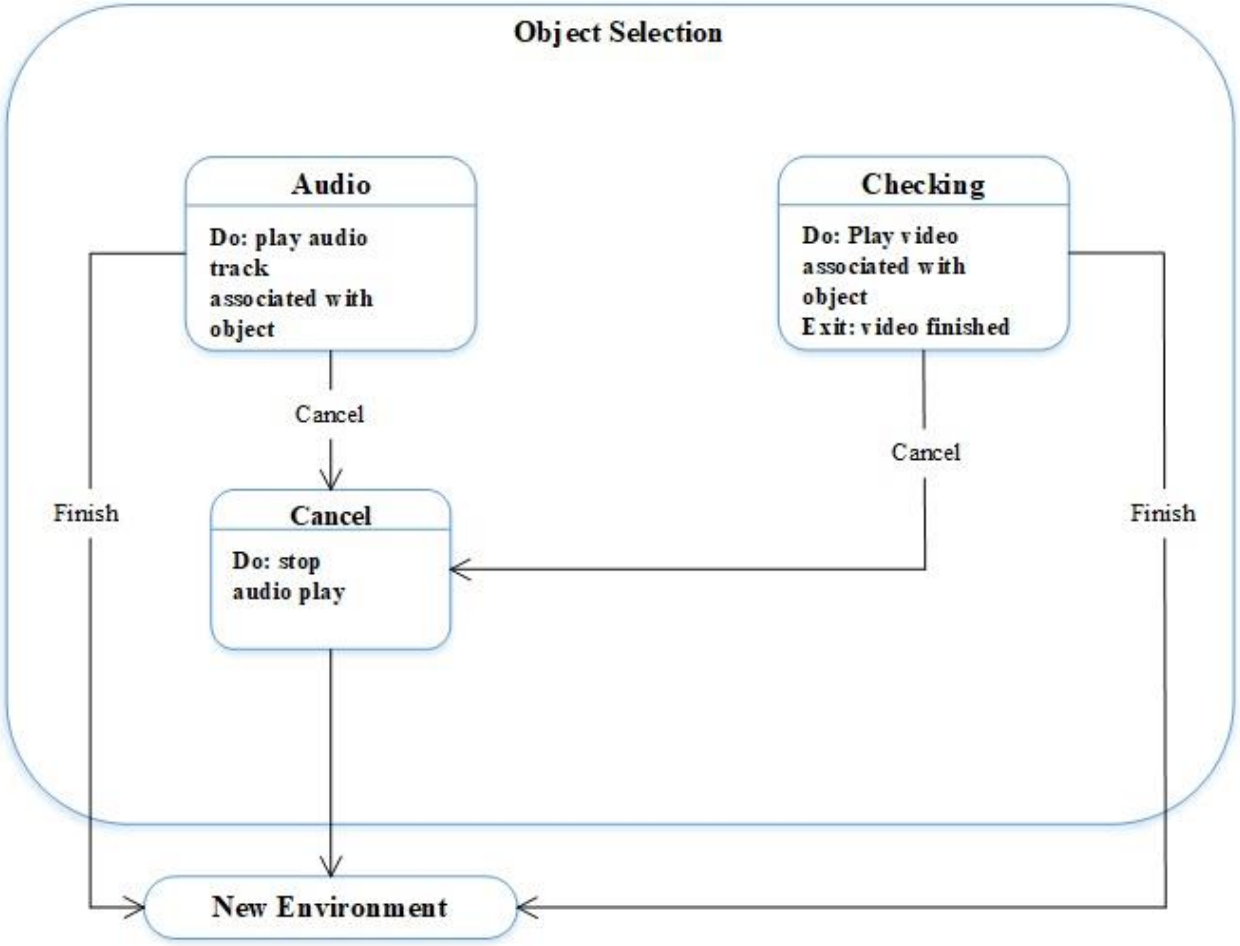
3.1 State Diagram

This diagram shows the flow of the system from one end to another [1]. In specific, activities from one state may lead to a following state depending on the conditions achieved. The arrows indicate the executed condition and direct to the next state.


Operation
Save edited environment to environment list
Save
Edit Environment
Do: load selected scene with editable objects and display buttons "Save" and "Exit"
Edit Environment
Exit
Object Selection
Do: perform action on selected objects
Sub-diagram for object selection
View/Edit Environment Page
Do: display saved environments in categories
Exit: selected environment
View Environment
View Environment
Do: display selected environment with clickable objects
Exit
Uploaded Panorama
Photo Album
Do: direct to iPhone photo album
Exit: Selected panorama
Camera
Take panorama
Exit: Action completed
Upload Panorama
Take Panorama
Uploaded Panorama
View/Edit Environment
Create environment (step 1)
Do: display options "upload panorama", "take panorama" and "exit" button
Create environment (step 2)
Do: load canvas and display objects for selection and display "Back" and "Next"
Exit
Another sub-diagram for verification
Create Environment
Next
Back
Login Screen
Do: display login & create account
Exit: action selected
Login
Home Screen
Do: display "View/edit environment", "Create environment", "Logout", "About", and "Settings"
Operation
Do: add new environment to environment list
Create environment (step 3)
Do: load completed scene draft and display "Back" and "Next"
Logout
Next
Exit
About
Done
Create environment (step 4)
Do: display field "environment name" and buttons "Back" and "Done"
Create Account
Submit
Back
About
Do: display copy rights of app and developer messages
Settings
Another sub-diagram for verification
Create New Account Page
Do: display copy rights of app and developer messages
Off
Do: turn off panning motion
Off
Settings
Do: display toggle panning motion button
On
On
Do: turn on panning motion

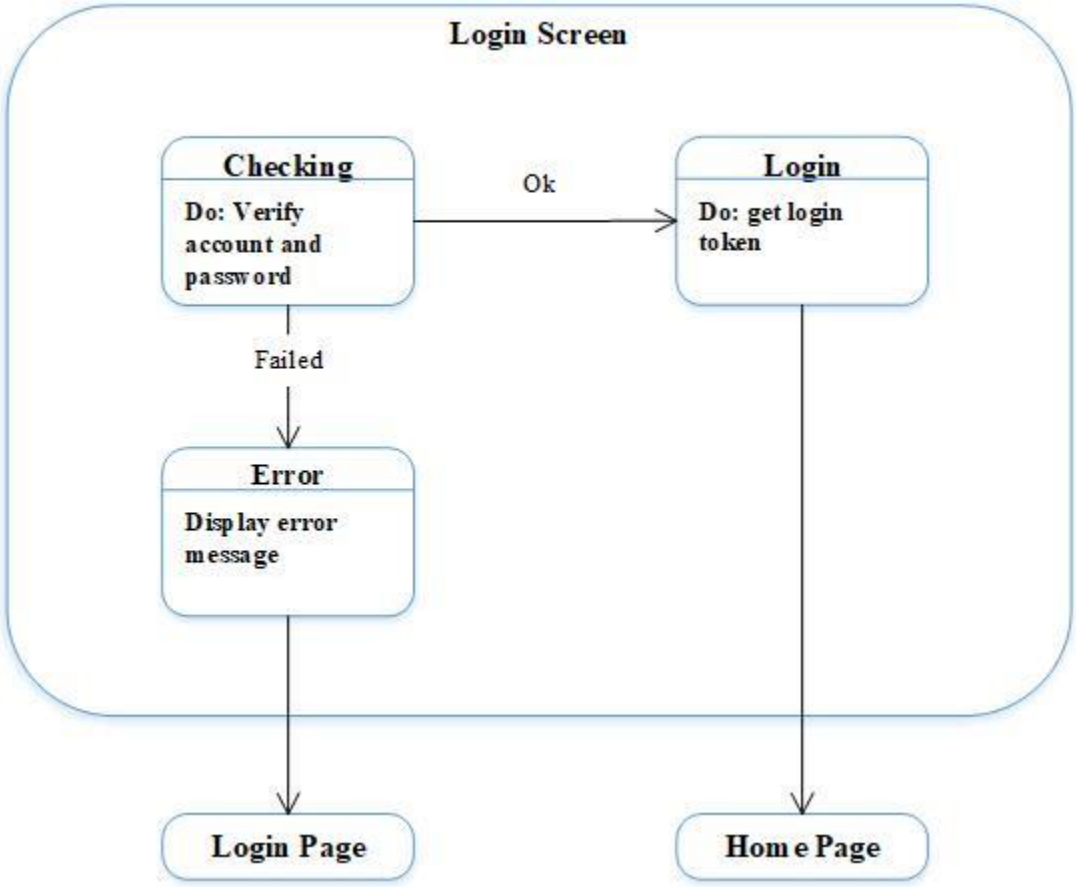
3.2 Sub-diagram for Object Selection

This is an expanded superstate, *Object Selection*.



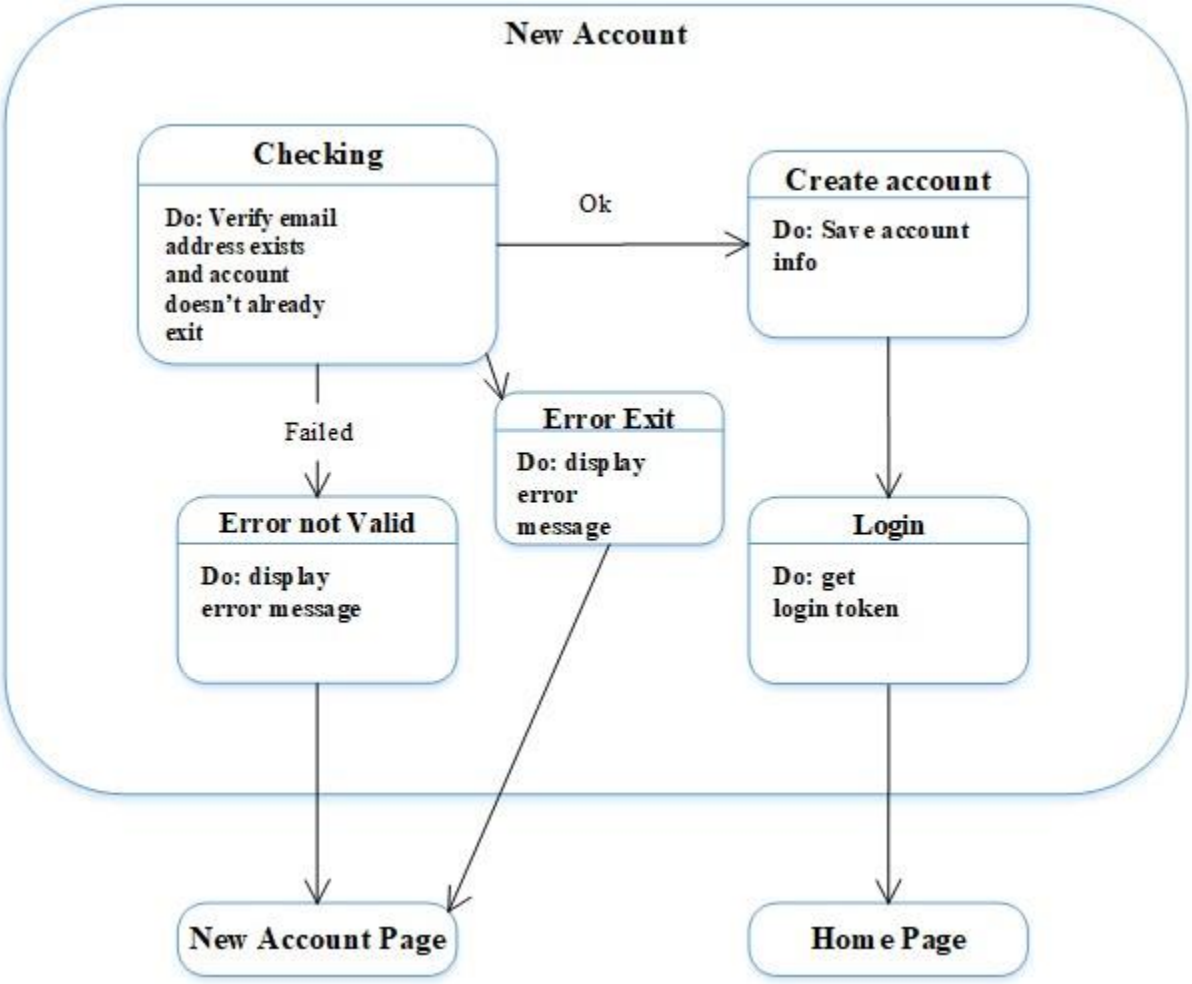
3.3 Sub-diagram for Login Screen

This is an expanded superstate, *Login Screen*.



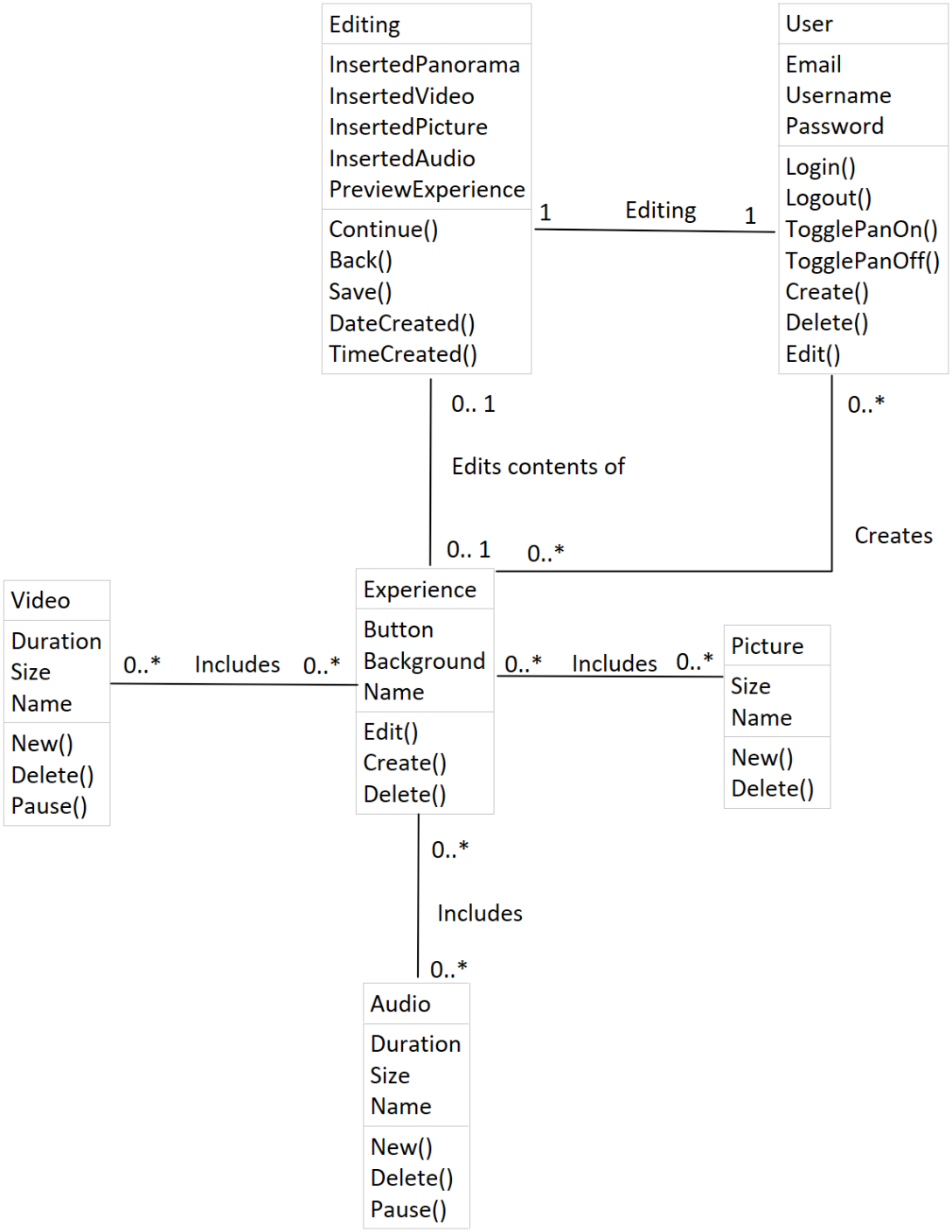
3.4 Sub-diagram for Create New Account Page

This is an expanded superstate, *Create New Account Page*.



4 Class Diagram

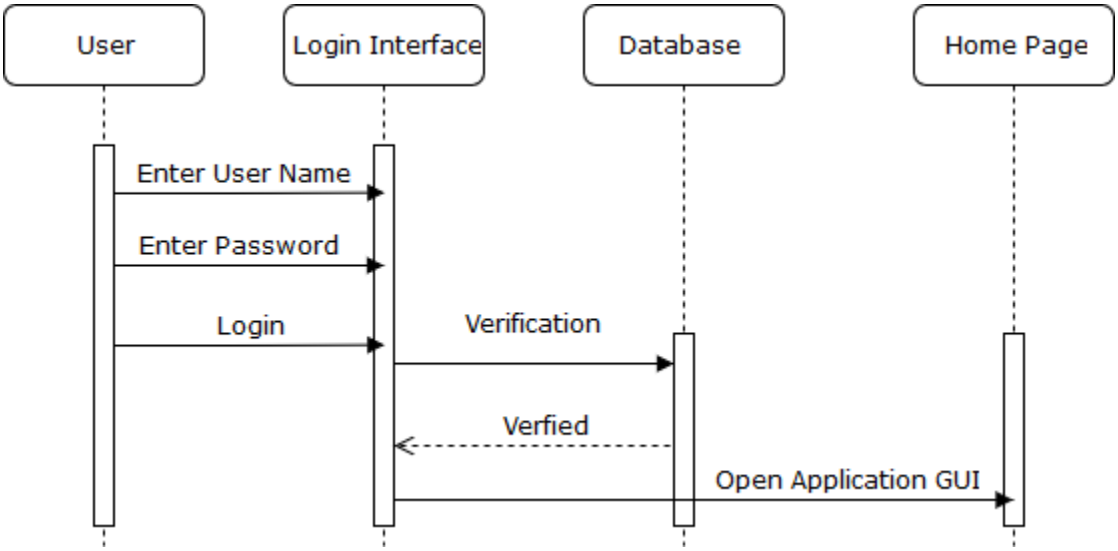
This is a class diagram for the system indicating dependencies and interactions between each class. Within each class, there are also functions and variables that make up its structure.



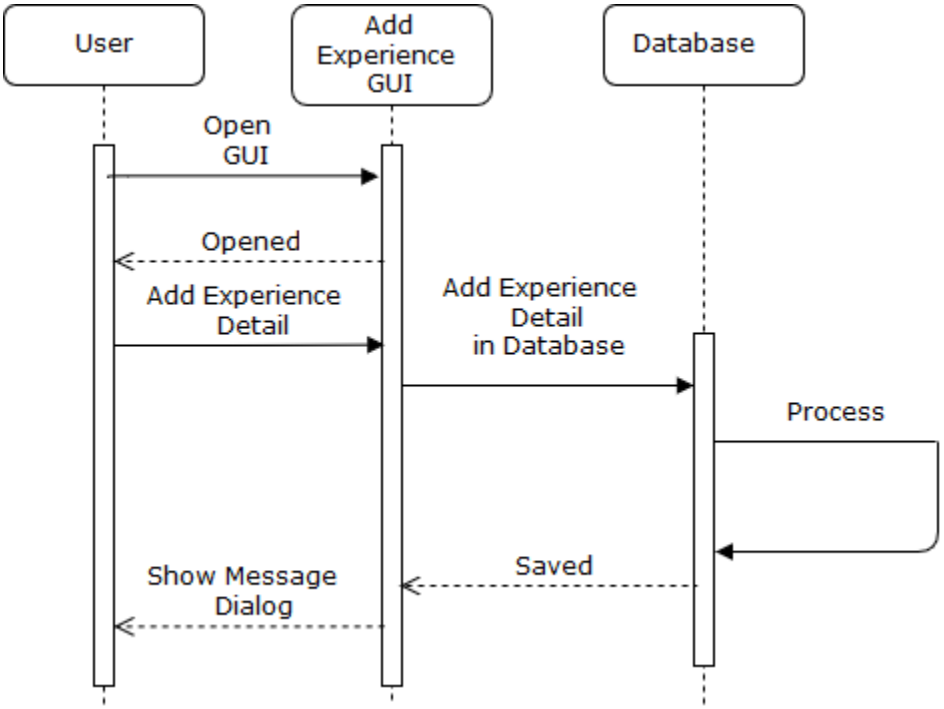
5 Sequence Diagrams

Each of these sequence diagrams show the output(s) of the corresponding input(s) [1].

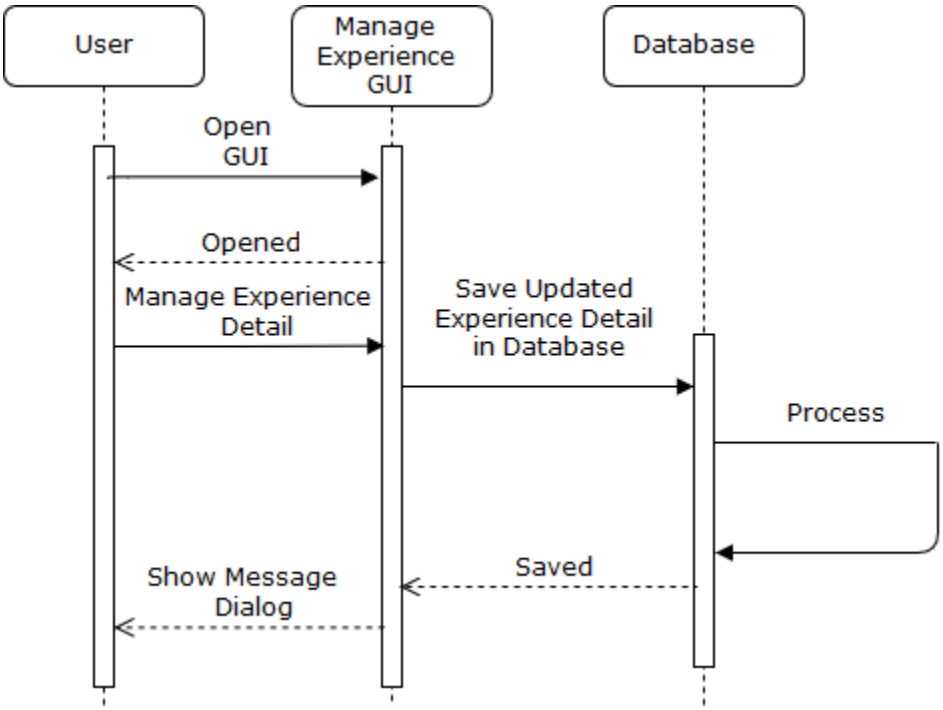
5.1 Login



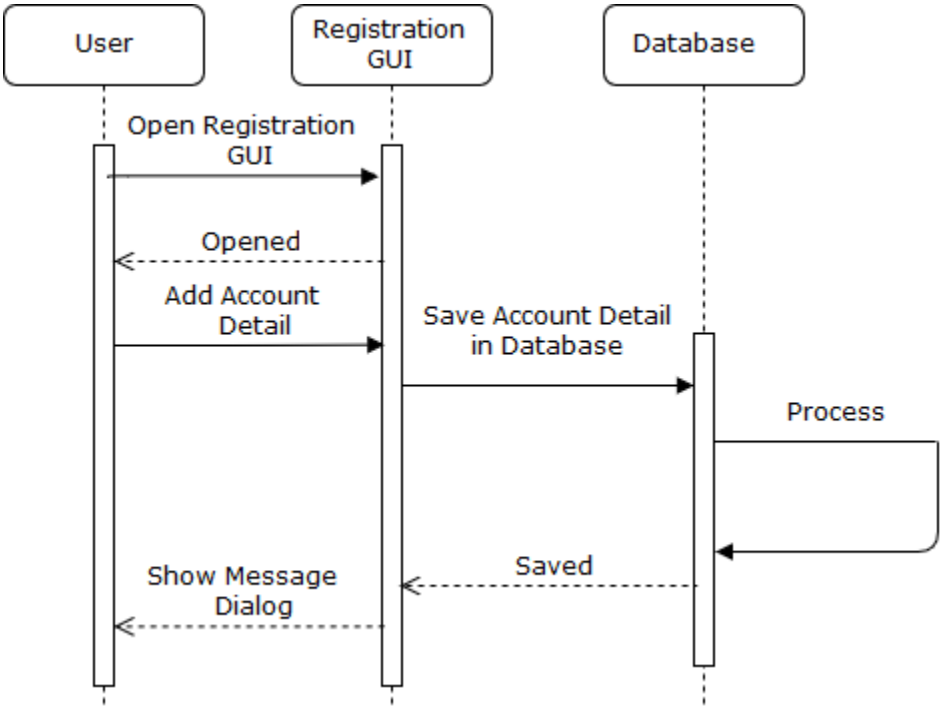
5.2 Create Experience



5.3 Manage Experience



5.4 User Registration



6 Data Requirements

6.1 User I/O

6.1.1 Homepage

User will use the IOS software keyboard to enter their username and password in the login screen. If the user-input login information is valid, the user will be directed to the home page. Files in JSON format are retrieved from the database after user has logged in. Then *experience* associated with the user is displayed on home screen. The app will first load the preview image and text description for each *experience* to populate each container shown in the home page, follow by sending queries to retrieve image, audio and video files in the background.

6.1.2 Editor page

Using the touch screen as an input device, the user is able tap on each button to navigate within the app, swipe left or right to view various parts of the scene, and use drag-and-drop motions to insert any predefined audio, video and pictures. When prompted to input the name of the experience file, user will use the IOS keyboards to populate the text field.

6.2 Database format

The app will use Firebase as the database for storing all the pictures, videos, and audio files. Audio files are stored in .mp3 format, video in .mp4 format, and pictures in .png format. When the iOS app makes a HTTP request to the web server, it will return files necessary to load the screen in JSON format.

7 Feature Priority

7.1 High Priority: Deployed in Version 1

Function ID: F4 (selecting and viewing premade experiences)
Function ID: F7 (app settings)
Function ID: F8 (motion control and swipe navigation)
Function ID: F9 (exiting an experience)
Function ID: F10 (sound/video/next panorama buttons)
Function ID: F11 (play/pause instance)
Function ID: F12 (video popup)
Function ID: F13 (go to next panorama in experience)
Function ID: F14 (warning if not completed)
Function ID: F15 (end of experience congratulatory message)

7.2 Moderate Priority: Deployed in Version 2

Function ID: F5 (add new experience/ edit panorama)
Function ID: F16 (edit title of custom experience)
Function ID: F17 (add/edit description of custom experience)
Function ID: F19 (add new panorama/take a panorama picture)
Function ID: F20 (panorama creation)
Function ID: F21 (delete experience)
Function ID: F22 (exit and save from experience editor)
Function ID: F23 (navigate panorama)

7.3 Low Priority: Deployed in Version 3

Function ID: F1 (log in)
Function ID: F2 (sign up)
Function ID: F3 (user registration details)
Function ID: F6 (settings page/sign out)
Function ID: F18 (delete/edit/view custom experience)
Function ID: F24 (add sound/video/audio to panorama)
Function ID: F25 (choose previous recordings to add to panorama)
Function ID: F26 (adding another panorama in experience)
Function ID: F27 (add or delete icons in panorama, preview sound/video)
Function ID: F28 (exit/save changes to experience, and back to editor page)

8 References

[1] I. Sommerville, Software engineering. Boston: Pearson, 2011.