CS 4720 - S17 - Final Project

Device Name: Kiku Platform: iOS

Name: Austin Luk Computing ID: al2gq

App Name: Dream Chronicle

Project Description:

This app, Dream Chronicle, is intended to serve as a combination dream journal and dream organizer, with an emphasis on assisting in the goals of recall and lucidity. As an avid dreamer (and recorder thereof) myself, I've seen and used many helpful tools that aim to help the user achieve these goals, but I have yet to see most of them in one place, and as such I thought it might be helpful to create something that does so. I hope for this app to be of assistance to anyone looking to obtain maximum enjoyment from the nightly escapades we far too often simply let slip from our minds.

This app does the following:

-The system shall allow users to log in under a created user profile that will allow them to access their dream journal from anywhere.

-The system shall allow users to type up a dream/dreams in the primary page and save them to a database.

-The system shall allow small drawings to be made and appended to dreams for later.

-The system shall be able to play audio from a pre-defined playlist while the user is asleep, assisting in the obtaining of lucidity (by way of binaural beats), getting to sleep (ASMR, meditation tracks), or both. Shaking the system will cease audio playback.

I plan(ned) to incorporate the following features:

-Build and consume your own web service using a third-party platform: Firebase will be used to allow users to store their dreams.

-Audio: Audio files will be playable while the user is asleep, and voice recording will be possible.

-Device Shake: A device shake will be used as a signal to stop audio.

-Local database storage: A version of the dream journal will be (potentially) stored in a local database.

-Local preferences: Storage of user name and other personal details for display.

-Open shared activity: Sharing of dreams through firebase or text message will be possible.

Sadly, not all of these were implemented, due to lack of time.

Wireframe Description:

This wireframe is intended to show the basic layout of Dream Chronicle application. There is a main page, where one may sign in or create an account. One a user is logged in, they may access one of three main pages through the “Record Dream”, “Dream List”, and “Sleep Playlist” buttons.

-Record Dream accesses a new page pre-populated with the current date, and presents a blank text box (which will expand and scroll upon being filled), as well as the options to record an audio version of the dream, append a picture, or save and return to the menu. The audio recording option will cause a pop up to appear for recording, while the drawing option will bring up a separate page of blank canvas to doodle upon and save. Saving the dream will bring up an option to “tag” the dream with 0 or many tags, which will appear in the Dream List menu.

-Dream List brings up a reverse chronological order list of previous dreams, with tags given to them visible to the right. A dream can be tapped on to bring up that dream and any associated drawings/audio, and edit/tag it further if so desired. An option to filter by tags is also available at the bottom of the screen.

-Sleep Playlist brings up a pre-loaded playlist of tracks designed to assist in obtaining sleep, obtaining lucidity in one's dreams, or both. It is designed to be used with the user wearing earphone or headphones while they sleep. Beside each track is a checkbox that can be used to indicate the user wants the track on the playlist. There is an option to turn off this feature at entirely at the bottom of the screen.

(The actual app is somewhat different than the original wireframe, but follows the same basic ideas.)

Platform Justification:

To be perfectly honest, this app would be perfectly at home on either platform. It not complex, does not charge money, and is not particularly intended to be widely used. However, as I did not have a partner and was more familiar with coding for iOS than Android, this was the natural choice. Truth be told, this section seems rather arbitrary for an app whose end goal is largely the functionality of the app itself, rather than any sort of internal company use or drawing in revenue. (Apologies if this is not an acceptable answer.)

Major Features/Screens:

-Login Screen: A simple view allowing the user to log in or create an account from any phone, retaining the ability to access their dream journal anywhere, from any phone with the app installed.

-Record Dream: The page that allows users to actually type out their dreams. The text box will scroll infinitely, allowing for dreams of any length to be written down.

-Add Art: A page which can be accessed from the Record Dream view, allowing users to create art depicting the related dream and append it to be viewed at any time. Has access to a variety of colors and an eraser.

-Dream List: The actual “journal”, which contains all of the user’s dreams and the date/time they were originally saved, displayed in descending order from most to least recent. Through this, any previous dream and its associated art can be edited and re-saved as the user desires.

-Sleep Playlist: A playlist of tracks intended to assist the users in either getting to sleep, or experiencing vivid/lucid dreams while asleep. Meant to be used with earphones, and includes ASMR and meditation tracks. Playback can be halted or resumed with a shake of the phone.

Optional Features:

Due to unique personal circumstances, I was only required to have roughly 40-45 points total for optional features, and was allowed to lock my app’s rotation (ask Prof. Sherriff for details).

-Device Shake (10): Shaking of the device while audio is playing will pause playback if it is playing, and resume it if it is not.

-Audio Management (15): Sleep Playlist contains a list of audio tracks that can be played, stopped, paused, rewound, and fast forwarded.

-Build/Consume Your Own Web Service Using a 3rd Party Platform (15): Firebase is being used to store all usernames, passwords, dreams, and art.

-Image Editing (15): Add Art allows editing/drawing and saving of images (point value assigned by Professor Sherriff).

-User Preferences (10, possibly): I am unclear if this counts or not, but users are able to log in via a password and access their data from anywhere. This is not necessarily separate from Firebase, but as I am unclear about the distinction I am including this anyways.

Total: 55-65 Points (at least 10 over personal required amount)

Testing Methodologies:

A full run-through of all features was tested for multiple users (Username, an established user with many dreams recorded, and Flairina, an initially empty account). The following problems were encountered during testing:

-Within the login view, attempts were made to log in as a nonexistent user, to log in with an incorrect password, and to create a user that already exists. These attempts failed. However, attempting to logout when not logged in did NOT fail due to an unnamed identifier. This issue was subsequently fixed.

-Attempts were made to “break” the application. Issues with dream list (an occasional random crash linked to loading times), add art (a strange scrolling that occurred whenever a point was touched on the canvas, but not moved), and record dream (art would disappear if not completed in one sitting) were subsequently fixed. The ability to draw on top of certain parts of add art was left in, with the reasoning that it allows for more canvas space if absolutely necessary, and that it does not prevent any buttons from working.

-An error occurred if record dream attempted to save text and/or art while there was no user logged in. To prevent this from happening in the future, the main menu buttons were locked while there is no current user.

-The database was changed manually, and checked for functionality with the app afterwards. All remained well as long as art images (which were stored as strings) were not edited. As this should not be a problem to begin with, it was considered unimportant to prevent and was ignored.

Beyond the above issues, no other problems were encountered during testing.

Usage:

While it is perfectly feasible to simply create a new user for one’s personal use, the current usernames and passwords in the database include the following:

-Username: Password
-Flairina: Password
-Teacher: UseMyApp
-Sherriff: PleaseGiveMeAnA

Username is currently populated solely with test cases, Sheriff contains one dream and one art piece, Flairina contains some copies of my own dreams (no art), and Teacher contains nothing.

Lessons Learned:

-The general workings of Xcode and Swift. Not that the lack of Swift 3 tutorials, and the ocean of Swift 2 tutorials I kept finding instead, made this all that easy.

-What needs to be accomplished by the storyboard vs what needs to be accomplished by the code. In Xcode, this generally boils down to “if it’s not a view, button, or segue? You need code”.

-Do not segue from every view to another- use dismissals instead, to save on memory and not make an ever growing stack. This may make things more difficult in some ways, but is far more efficient and easier on memory.

-Similarly, how to use segues, dismissals, and delegates. It was a long, arduous struggle to figure out the latter once I found out dismissals did not qualify as segues, and thus did not allow for the passing or reloading of data on previous screens.

-How to create, use, and edit an online database. Firebase is regrettably not as easy as one would hope, despite their many tutorials, but it is possible to learn, at least. Firebase STORAGE however, I couldn’t tell you anything about- hence why all my images are stored as strings.

-You should test your application CONSTANTLY. While this is somewhat common sense for coding in general, in mobile development there are additional complications in that the problem can lie not just in one’s code, but in storyboard and interface issues. As any one thing can screw up other things, making sure what you have up until a certain point is crucially important.

-In the same vein, build the application to the intended device as often as possible. The end application is not going to be run on a simulation, but on a physical mobile device. Thus it makes far more sense to do one’s testing on said mobile device, to make sure ahead of time that there are no issues transferring from code to phone.

-Mobile application development should most DEFINITELY be done with a team. Even for small applications, doing the entire thing yourself is a recipe for stress, problems, and staying up until 5 in the morning.