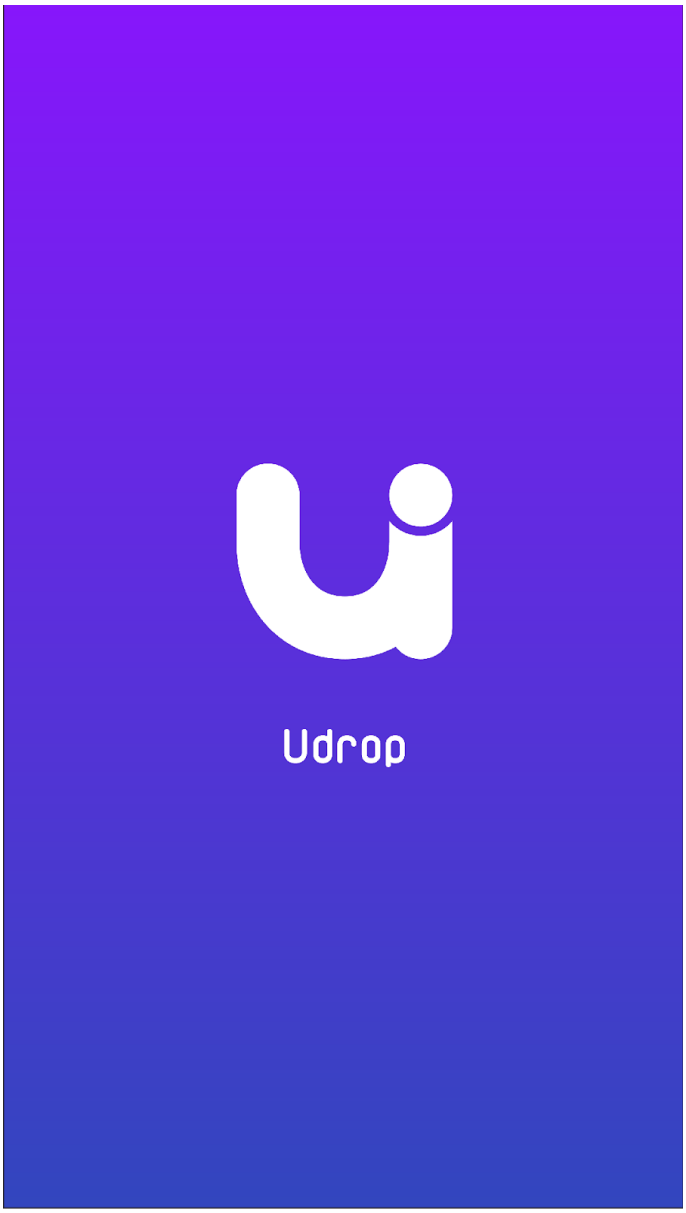
Udrop



6/10/20

CEO: Dr.A.I Concepcion

Prepared By:

Francys Cunanan

Hanane Elkarni

Section 1: File Structures

The team uses Git's terminal to manage and keep our file structure. Originally, the files are intended to be accessed through Bitbucket. However, the dev team stored the files in a GitHub repository instead so that it is more convenient to use. All the group members that participated in the project have access to it and can upload their own builds without merging into someone else's.

LoginActivity.java

This activity will be the starter page that the user will be sent to. For any existing users, this is a gate needing access in order to have access to Udrop's services. The user may choose to select save login information to avoid needing to input their email and password every time they launch the app. If the user does not have an account, the user can be redirected to make an account by pressing the sign up text displayed on the interface.

SignupActivity.java

This activity provides multiple text fields for the user to fill in. They are pieces of information required to help build the user's account.

HelpActivity.java

This activity provides the user with information so that it may assist them to understand the app better. The app is straightforward and easy to understand, but for convenience sake if the user still does not understand how the app works, there is an explanation function that will discuss it briefly on the most recent prototype. Future plans to help make it stand out is perhaps display a question mark symbol that can explain everything from there.

FriendsActivity.java

Upon logging in, the user is brought to this page. Here the user may view their friends list, send friend requests, remove friends, or select a friend to message. There is also a menu bar that allows the user to navigate to other parts of Udrop, like Drops, Profile, and Places.

DropsActivity.java

Here the user can view drops (messages) that they have received from their friends. If a drop has a red circle, it means that the user cannot view the message because they have not visited the drop location yet. If a drop has a green circle, then the user has visited the drop location and may view the drop message. The user may also delete drops. There is also a menu bar that allows the user to navigate to other parts of Udrop, like Friends, Profile, and Places.

PlacesActivity.java

Here the user may view their saved places, add new saved places, or remove places on their list. To add a new place the user first must input the name like Home, School, or CSUSB Parking Lot F, then click the new location button. There is also a menu bar that allows the user to navigate to other parts of Udrop, like Drops, Profile, and Friends. Upon clicking the new location button a google maps view will open and the user can search for the location they desire manually, or by using the search bar. They have the option to save the place, or cancel. .

MessengerActivity.java

The user is brought here upon clicking on a friend from their friends list. Here the user will enter a subject, message, and select a location from their saved places, or add a new location to use. If they decide to add a new location, they will be brought to a google maps view where they choose the new location in a similar fashion to the Places Activity. If the user wants to save the location, then they hit the save icon, otherwise they can hit the send icon to use the selected location for the message, but not save it.

ProfileActivity.java Here the user may change their profile picture, nickname, email, or phone number associated with their account. There is also a menu bar that allows the user to navigate to other parts of Udrop, like Drops, Friends, and Places. They can also logout from this activity.

Section 2: Instructions

Android:

1. You must have access to the github repository to download the application files.

2. Once downloaded, extract the files to a directory of your choosing.

3. Start Android Studio and under the toolbar option File select File => Open and open the directory you stored the application files in. Inside the directory, choose the folder titled "Udrop_Prototype_2".

4. Check the gradle files and update any implementations that are out of date. Afterwards Sync the gradle files with the project by clicking the little elephant icon in the top right of Android Studio.

5. To run the app on a physical Android phone, you can google how to set up a phone for debugging on Android Studio. Otherwise, you will need to setup an emulator. There are plenty of tutorials for setting up an emulator online. Note: the app will run on emulators/phones with API level 25 or above. If your phone is no more than 5 years old, it should work just fine.

6. Once your physical phone is connected and set up, or your emulator is built, click the play icon (little green play arrow icon in the top right of Android Studio) to install the app.

7. The app should automatically launch once your phone/emulator is unlocked and the app is fully installed. If not, then open the app manually from the phone/emulator home Screen.

iOS:

1. You must have access to the github repository to download the application files.

2. Once downloaded, extract the files to a directory of your choosing.

3. Install the app dependencies by opening up a terminal window, locating the Podfile file under the "Final_Udrop" folder and running the following commands:

```
cd ~/path_to_downloaded_folder/final_udrop
pod install
```

4. After running "pod install", an *.xcworkspace* file will be generated in the same folder. Double click on it to open the entire Swift project in Xcode.

5. Choose which simulator you want to run or plug in your iPhone device and click on "Run" to install the App on the chosen device.

Section 3: Good Implementations

The following are positive outlooks and benefits to the current state of the app:

- The app offers familiarity so that the user can be up to speed on how to use the app.
- The app also has its own design, functionality, and visual flair that distinguishes itself from the other already existing social media platforms.
- The interface's simplistic layout allows quick response times from the app.
- The app has access to both platforms: iOS and Android.
- Most of the difficult features have already been accomplished on the prototype, which will prove beneficial for the next team to look towards its future.

Section 4: Needs Improvements

The following are weak points of the current implementation:

- The app is marketed to be primarily used by iOS users, but currently the iOS platform for the app has not gained progression as fast as the Android counterpart.
- The app's database is limited to Firebase from Google.

The very weak points will be addressed to the next team that are assigned to continue the project.

Overall Recommendations:

In retrospect, the following will suggest new features and recommendations that will help improve the app's development and quality of life features:

- Tutorial for the new user on how to set up a Udrop message to a specific location, who can receive it, and set its expiration date.
- A more eye-popping interface that follows the source code's color scheme yet also specifically made to attract youthful college students into using it.
- Find a bigger, stronger database so that the app can be tested on bigger numbers.
- Future teams should be able to communicate with their members more frequently so that people can be on task and work together more accordingly and efficiently. This is a major issue when working together online.
- Implement a proper life cycle to each activity in the Android version. Currently there are only onCreate and onDestroy functions for each activity, but the full range of the life cycle for each activity needs to be implemented for optimal app performance and stability.