SOCIAL MOBILE APPLICATION: UDROP

Mahmoud Oraiqat

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd

Part of the Science and Technology Studies Commons

Recommended Citation
https://scholarworks.lib.csusb.edu/etd/1539

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
SOCIAL MOBILE APPLICATION: UDROP

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Mahmoud Oraiqat
August 2022
SOCIAL MOBILE APPLICATION: UDROP

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Mahmoud Oraiqat
August 2022
Approved by:

Bilal Khan, Advisor, Computer Science

Jennifer Jin, Committee Member

Haiyan Qiao, Committee Member
ABSTRACT

Udrop is meant to be a global platform to seamlessly deliver content by simply “dropping” content at designated locations for select users. Users can drop content in the form of texts, short videos, and pictures, which users will then be able to share with their friends. Udrop features will provide an interactive user experience for users to socialize and expand their social circles. Udrop ensures scalability, reliability, and a manageable innovative social application by using cloud services. Currently Udrop is in a beta-trial with 10 to 20 users generating anywhere from 5 to 15 records, drops, per user. Udrop's end-user interface and internal system design ensure that user requirements such as low latency, clear graphical design, and reliable service are addressed. Udrop will provide everyday social media users a new take on how their social media can provide a more meaningful interaction with their audience.
TABLE OF CONTENTS

ABSTRACT .................................................................................................................................................. iii

LIST OF FIGURES ...................................................................................................................................... vi

CHAPTER ONE: INTRODUCTION ............................................................................................................. 1

  Background ................................................................................................................................................ 1
  Significance ............................................................................................................................................... 1
  Purpose .................................................................................................................................................... 2
  Scope ...................................................................................................................................................... 3

CHAPTER TWO: OVERALL DESCRIPTION ................................................................................................ 4

  Product Perspective .................................................................................................................................. 4
    User Interfaces ....................................................................................................................................... 4
    Cloud Services ....................................................................................................................................... 5
    Communication Interfaces ...................................................................................................................... 5
    Memory .................................................................................................................................................. 5
    Operation ............................................................................................................................................. 6
    Security ................................................................................................................................................ 6

  Product Functions .................................................................................................................................... 6

  Users ....................................................................................................................................................... 8

  Constraints ............................................................................................................................................. 9

  Class & Use Case Diagrams .................................................................................................................... 9

  Specific Requirements ............................................................................................................................ 11

  User Interfaces ...................................................................................................................................... 12
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Interfaces</td>
<td>13</td>
</tr>
<tr>
<td>Function Requirements</td>
<td>14</td>
</tr>
<tr>
<td>Performance Requirements</td>
<td>16</td>
</tr>
<tr>
<td>Design Constraints</td>
<td>16</td>
</tr>
<tr>
<td>Software System Attributes</td>
<td>16</td>
</tr>
<tr>
<td>Assumptions and Dependencies</td>
<td>16</td>
</tr>
<tr>
<td>CHAPTER THREE: USER TEST CASES</td>
<td>17</td>
</tr>
<tr>
<td>Test Case 1: User Authentication</td>
<td>17</td>
</tr>
<tr>
<td>Test Case 2: Interacting with Friends</td>
<td>21</td>
</tr>
<tr>
<td>Test Case 3: Sending a Drop to Users</td>
<td>25</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>34</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

Table 1.1  Udrop Client Flow Chart Between Cloude Services. ............................ 7
Table 1.2 Udrop Interface Flow Map. ........................................................................ 8
Table 1.3 Class Diagram. ......................................................................................... 9
Table 1.4 UML Use Case Diagram ........................................................................... 10
Table 2.1 Login Page And Create Account Page ..................................................... 19
Table 3.1 Mongo DB Users Portal ........................................................................... 20
Table 3.2 User Inputting Additional Information ..................................................... 21
Table 3.3 Home And Friends Page .......................................................................... 23
Table 3.4 Change In Friend Status ........................................................................... 24
Table 3.5 Actions For User Jeffry ............................................................................ 26
Table 3.6 Drop Creation Page .................................................................................. 27
Table 3.7 Drop Media Page After Providing A Custom Location .............................. 29
Table 3.8 Home Page After Sending A Drop To User Jeffry ........................................ 30
Table 3.9 Notification And Home (Locked) Page ...................................................... 31
Table 3.10 Drop Location And Home (Unlocked) Page ............................................ 32
Table 3.11 Comments Page ...................................................................................... 33
CHAPTER ONE
INTRODUCTION

Background

Social media apps are being published every day; however, the majority seem to branch further and further away from reality. This is in a large part due to how enticing socializing has become through technology. Popular apps such as Facebook, Instagram, or Twitter allows user to send and receive their social media with the push of a button. Although users are guaranteed convenience, these applications are depriving users from physical interaction. Udrop attempts to bridge the gap between the virtual and real world by essentially allowing their users to send virtual care packages within the real world for the users to retrieve. This will allow users and their friends to have a more meaningful interaction with one another, an interaction that will not end at the press of a button. Udrop allows users to receive messages when and where they need them.

Significance

Udrop will be a new platform for everyday app users to have a more meaningful interaction with their audience. This will be achieved by allowing user to receive message when and where they need it.

Although Udrop is categorized as a social media platform, there is no limitation to a user’s creativity when creating a drop. Drops can used as the
medium to make their everyday social media post more meaningful for them and their audience.

Udrop will one day be published on the Google Play and App Store for distribution.

Purpose

Udrop is being developed for users that want more from their social media apps. Udrop will allow users to drop messages in the form of text, images, or short videos to recipients at designated locations to be collected. Depending on the designated location a user specifies, users can provide a meaningful interaction with their media drop. For example, say a user walked into a grocery store, they check their phone, and they notice they unlocked a Udrop message. The message contains a grocery list that their significant other had verbally mentioned to them that morning. However due to a long workday this user forgot everything they were told. Fortunately, their significant other sent them a drop with the grocery list at the store’s location in case they forgot when they arrived at the store. Now, said user can send a confident message back in the form of a comment, notifying their significant other mission complete. In this scenario the social media message had a significant role in a trip to a grocery store.

The objective of this project is to build an application prototype by implementing the required features requested by the client and ensuring
scalability. This document will define the app’s functionalities, limitations and requirements provided by the client.

Scope

Udrop’s user interface (front-end) includes the following well-standard pages: Home, Friends, Account, Drop Location View, Create Drop, Login, Create Account, Notifications, and splash pages. The pages will be functional, easy to navigate, and provide the same aesthetic in both iOS and Android platforms. Udrop will use cloud service technology to handle all server (back-end) functionality to manage user data and client connections. The main objective of this project is to provide a working, scalable, prototype where all application features and functions are implemented.

As Udrop is a social media platform, Udrop’s design will encompass a way for users to expand their network. Udrop will enhance social interactions by connecting people through a mutual friends drop. Allowing users to share their ideas and display names through a drop’s comment section.
CHAPTER TWO

OVERALL DESCRIPTION

Product Perspective

Udrop is a standalone application that will provide users a social media service. Udrop will allow users to send and receive drops in the form of a message, image, or short video. These drops will be a user creation and will be sent to their designated receiver with a specified location.

The application is designed in a way where users have the option to send a drop to multiple users thus bringing people together at the drop location. Once the drop is received users are then able to provide feedback by leaving a comment that all user who received the same drop will be able to view.

To get desired functionality, Udrop will have one main component in the form of a client which will connect to multiple cloud services. The client will be responsible for all user interface components and communication between the cloud services through SDKs. The cloud services will then manage data, client connections, and user requests by using their respective development tools/services.

User Interfaces

Udrop’s user interface (front-end) includes the following pages: Home, Friends, Account, Drop Location View, Create Drop, Login, Create Account, Notifications, and splash pages. The pages will be functional, easy to navigate, and provide the same aesthetic in both iOS and Android platforms. Within
selected pages, the client will communicate with cloud services to dynamically update specific viewing components with the latest user data.

Cloud Services

There are four main cloud services that will be used for the development of Udrop:

1. React Native IDE (Client-Side development)
2. MongoDB’s Realm (NoSQL, and Serverless Architecture),
3. AWS (media storage using S3 Simple Storage)
4. Google Maps API (google maps service).

Communication Interfaces

Communication between services will be handled through embedded SDKs within the React Native client to handle different functionality. The client will then manipulate data between the services to provide a cohesive system between SDK functionalities.

Memory

Udrop will use physical resources provided by cloud services MongoDB, and AWS S3. The amount of memory used will follow cloud services storage model where Udrop's available storage will scale in accordance with the amount used.

On physical devices, Udrop will require 90 MB in storage space to download the app. Udrop will be able to run on any smart device with Android or IOS operating system.
Operation

Udrop implements Serverless Architecture where all underlying infrastructure is being handled by the cloud services. MongoDB’s Realm provides serverless computing using functions, saving data to MongoDB’s Atlas (NoSQL solution), and provides user authentication. While AWS S3 will store pictures and videos that can be readily accessed through URL links. By using these cloud services Udrop can ensure cost efficiency, elastic scalability, rapid iteration, low administrative overhead, and developer productivity.

Security

Since users will be meeting within a physical location security must be addressed accordingly so that users won’t be put into risky situations. Although users are responsible for adding Friends they trust, Udrop may provide a way to authenticate identity before accepting any drops from users they may not fully trust. A potential solution would be in the form of a text-message authentication or email authentication. Once users authenticate one another they will be allowed to send drops to one another.

Product Functions

Udrop follows a microservices model where the different cloud services are independent from one another as shown in Figure 2.1. The user interface will provide multiple pages that the user can navigate between to access different
features of Udrop.

Figure 2.1. Udrop Client Flow Chart Between Cloud Services.

An overview of how a user will interact with Udrop’s interface is show in Figure 2.2. Majority of the arrows shown below are double sided indicating that a user will be able to go back and forth between pages. However, there are some cases where the client determines which page to present to the user based on their user data. For example, if the user’s data confirms that the user signed in with a new account, they will be automatically navigated to Account Setup.
The general users of Udrop are individuals who would like to connect with friends, family, fans, followers, or likeminded individuals. Udrop is designed to provide a fresh take on social media interaction and will be available to any individual with a smart device. The target audience are those that would like their discourse to be more involved with the content they share. Similarly, Udrop is meant for users that would like to be proactive in receiving new content from those that they admire most such as friends, family, or even celebrities. It should be mentioned that there is currently no restriction on a user's age.
Constraints

- Udrop messages will be deleted after a 30-day period, which will be un retrievable to the user.
- Udrop is intended to be a free application but will include the use of google ads.
- Other constraints to this project are providing reliable scalability, cost efficiency, and a great user experience.

Class & Use Case Diagrams

The Class Diagram (Figure 2.3) provides a visualization of the relationships between Users, Friends, Drop messages, and Notifications classes. Users will typically have zero to many Friends, Notifications or Drop Messages depending on their activity. While Drops can have media content and may have feedback in the form of a “Heart” or a Comment.

![Figure 2.3 Class Diagram](image-url)
The Use-Case Diagram (Figure 2.4) shows the interactions between a Udrop’s system and environment.

1. **Realm System**: is responsible for registering users and Editing User Profiles
   
   a. **Register new account**: Realm System receives an account registration request and accepts/rejects after verifying submitted credentials.

   b. **Edit User Profile**: Realm System verify permission levels and update user information.
c. **User Validation**: Realm System will verify users and their permission level to allow access to data.

2. **User**: is responsible for dealing with all functionalities below.
   a. **Register with Udrop**: A user requires registration with verifiable credentials to use Udrop.
   b. **Add/Delete Friends**: Users can send friend requests or delete friends on Udrop.
   c. **Create a Drop**: Users with friends can create drops that are attached with media or text that will be retrieved at a specified location.
   d. **Locate Locked Drops**: Users who received a drop will be able to locate a drop to unlock it.
   e. **View Notifications**: Users can view friend requests or drop notifications.

3. **AWS S3 System**: is responsible for verifying profile or media data.

4. **Google Map API**: is responsible for validating all geographic locations.

**Specific Requirements**

All pages should provide a good user experience and reliable functionality. User’s experience is expected to increase with well-designed pages and interactive graphical components like buttons, scroll wheels, or dropdown menus. While functionality should be implemented such that all fail cases are handled and functionality is persistent.
User Interfaces

1 Login Page: Allows the user to sign into their account, then navigate to the user to the homepage. If the user does not already have an account, the user can navigate to the Create Account Page (Figure 3.1 Left).

2 Create Account Page: Allows new users to create an account by submitting an email, and password to be associated with their account. Once account is successfully created, user will be taken to Login Page (Figure 3.1 Right).

3 Welcome Page: Allow first time users to setup their account by providing a user icon, display name, and phone number. Once data is submitted successfully, users are directed to Preferred Location Page (Figure 3.3 Left).

4 Preferred Location Page: Allow users to provide a location they would prefer to go to access their drops. The preferred location can be changed at any time by navigating to the Account Page (Figure 3.3 Right).

5 Home Page: Allows the users to see their drops and navigate to Account or Friends page. There is also a floating button to navigate to Drop Creation Page at any time. Users can also navigate to a Drop’s comment page view comments (Figure 3.4 Left).

6 Account Page: Allows users to change their profile picture, update their preferred location, and log out. There is also a floating button to navigate to Drop Creation Page at any time (Figure 3.9).
7 Friends Page: Allows users to view friends, add friends, delete friends, and select a friend to drop to. There is also a floating button to navigate to Drop Creation Page at any time (Figure 3.4 Right).

8 Drop Creation Page: Allows users to choose/modify their target user(s) and specify drop location from three options (Figure 3.7).

9 Add Media Page: Allow users to provide a message or media in the form of pictures or videos (Figure 3.8).

10 Drop Location Page: Allow users to toggle between their real time location and the drop location on a google map. Once users are within 30-meters, they will be directed to the Home Page to view the drop (Figure 3.11 Left).

11 Comments Page: Users can leave a comment for their sender or other individuals who are viewing the same drop (Figure 3.12).

12 Notification Page: Allows users to view their notification that are generated from receiving new drops or friend requests (Figure 3.10 Left).

Software Interfaces

Account and Media Page have access to the AWS S3 SDK to upload their user icon profile and any media they wish to attach to their drop. The return data from AWS S3 will be in the form of a URL link that will then be saved to MongoDB Atlas through the MongoDB Realm SDK. The React Native client will provide a realm instance where data will be synced between all client connections. This allows users to received real time updates when there is a relevant change to user data: receiving notifications and drops from other users.
Lastly, the React Native client leverages Google Map API to display locations, set location markers, and view the 30-meter range around a drop location for specific pages.

**Function Requirements**

1. **Authenticating Users:** Udrop will authenticate users by allowing them to create an account using an email and password. Once the account is created the user will then be able to setup their account by providing a profile image, display name, preferred drop location, and phone number. All data is saved in their respective unique user documents within the database.

2. **Choose a Drop Location:** Users will be able to send drops to one or more friends that are registered in their Friend’s List. Once the drop target(s) are identified, the user will then choose one of three options to indicate the drop location. Firstly, users have the option to send a drop to their target(s) preferred locations. Secondly, users can provide an address that they would like to drop to. Finally, they can select an exact location on a map and provide a custom description of their desired drop location. Also, as an added feature users will be able to see their recent drop locations and “favorite” their favorite drop locations.

3. **Adding Media to Drop:** Once users choose a drop location they are then able to add a message and/or media to their drops. The media can be in
the form of a picture or a 10 second video that they can either take in the moment or choose from their device’s storage.

4. **Add/Delete Friends:** Users will be able to add friends by providing their friend’s email or display name. Once the friend request is sent, they will see a relationship status of “pending” within the Friends Page. The recipient of the friend request will then decide whether to accept or deny the friend request. Once users become friends, each user has the power to unfriend one another. Any changes made to the relationship status within the friends list will be updated in real time for both users.

5. **Unlocking a Drop:** Once a user receives the drop, the user must be in the 30-meter range of the specified location to unlock the drop. Unlocking the drop allows the user to see the contents of the drop from the sender. Users can then “heart” or comment on the drop within the comments page.

6. **Change Preferred Location and Profile Image:** Users can navigate to the Account page to change their preferred location and profile image.

7. **Handling Notification:** Notifications are generated when a user receives a new drop or friend request. User can accept or deny the friend request only in the Notification Page. While the drop notification will navigate the user to the Drop Location Page to toggle between their location and the drop location. Once a notification is interacted with, they are then deleted.
Performance Requirements

Udrop will be available on the App Store and Google Play and expect to run on most versions of Apple iOS and Android. The backend must be able to handle many requests with low latency, scalable storage and efficient in-app data processing.

Design Constraints

Udrop will be designed in React Native for multi-platform support, so page components must be formatted reliably so that regardless of the device used the pages will look consistent.

Software System Attributes

Udrop messages can only be retrieved if the specified recipient is in the designated drop zone. Users should not be able to see what the user sent until the drop is unlocked.

Drops can only be sent to friends that have already accepted the friend request. User should not be able to see friends with a “pending” status within the dropdown menu within the Create Drop Page.

Assumptions and Dependencies

The assumption can be made that the user will provide Udrop access to the internet and geological location for the application to send/receive messages, and utilize different features as intended. In addition, users are expected to have a basic familiarity of enabling their GPS and finding desired locations on a map.
CHAPTER THREE

USER TEST CASES

Test Case 1: User Authentication

Test Case Objectives:

1. Udrop should be able to process new users by providing a simplistic and easy to follow account creation procedure.

2. Once an account is created Udrop’s client must determine whether the user is a first-time user or a returning user. If the user is new to the Udrop platform, allow user to fill in additional account information. Otherwise, if the user is a returning user, direct them directly to their home screen to view their drops.

3. On Udrop’s Realm Instance (MongoDB Cloud Service), once users are authenticated, the Realm instance must create an empty user object within the database ready for user data to be uploaded.

4. Users should not be able to create an account with an invalid email address or an empty password. Similarly, user should not be able to login with invalid credentials.

Expected Results: Udrop fulfilling all test case objectives and displaying all features in a visually pleasing and practical manner. There should not be any visual components leaving the user guessing on their respective function or purpose.
**Process and Actual Result:** Udrop will automatically be on the sign-In page when user open the app. Users are then able to locate and navigate to the create account page as shown in Figure 3.1. Users will then fill in their desired email and password they would like to associate with their account. Once users tap “Create Account”, the client will interact with MongoDB Realm SDK to create a new user with the email and password provided, while also creating a new user object. New users will now be added to the list of authenticated users as shown in Figure 3.2.
Figure 3.1: Login Page And Create Account Page
Once the Realm instance is successful, the client will then navigate back to user login, where user can sign into their new account. Udrop will complete new User creation by navigating users to the Welcome page where users can provide a display name, phone number, and choose a preferred location. Once additional formation is verified on the Realm Instance, user is navigated to the home page to continue their Udrop experience.

Users are presented an alter window describing errors in regard to their credentials if the following conditions are violated: Empty email or password field, account does not exist, or incorrect email/password.
Figure 3.3: User Inputting Additional Information. (Using A Fake Phone Number)

Test Case 2: Interacting with Friends

**Test Case Objectives:***

1. Users should be able to add or delete one another from their Friends page. This entails finding another user with the same email or display name provided by the client’s user.
2. Users should not be able to add one another if users are already friends or if the email/display-name provided is not found in the database.

3. If a recipient declines a friend request, this change should be conveyed to the user by delete the pending request from the user’s friend list.

4. Users should be able to delete any friend on their friends list. The Udrop must update both users of the change in their friends list after deletion is successful.

**Expected Result:** Udrop will be able to add users using an email or display name. Alert the user to any errors such as an invalid email or display name. Update Udrop’s user interface when requests are successful and there is a change in a user’s friend list.

**Process and Actual Result:** Users can begin adding friends by navigating to the Friends page from their Home page then tapping the “Add Friend” button located at the top right of the screen. A pop-up widget will allow users to input an email or display name to add (Figure 3.4). If the user is found, the pop-up widget will close, and the friends list will populate a new friend component displaying the desired user’s display name with a pending status. However, if the user is not found, the pop-up widget will display red text indicating the failed request, allowing users to try again. Once the recipient
accepted the friend requests both users will see the change within their Friends Page respectively (Figure 3.5).

Figure 3.4: Home And Friends Page
Users will also be able to delete a friend by tapping on the user’s friend component they wish to delete and press “Unfriend” on the pop-up widget (Figure 3.6). This will remove one another as friends between both users. Users would then have to request another friend request to become friends again.
Test Case 3: Sending a Drop to Users

**Test Case Objective:**

1. Users can send a social media drop to one or many of their friends.
2. Users can choose from three different options when specifying the drop location: their target(s) preferred location, a specific address, or a specific location where they provide a custom description of the drop zone.
3. Users can also reuse a recent drop location or a favorite location.
4. Users should not be able to proceed to the next step in drop creation without providing choosing recipients or a drop location.
5. Users will then be required to provide a message and given the option to attach a picture or 10s video.
6. Lastly the client should not direct the client until any media is saved within the database and is readily accessible.

**Expected Results:** Udrop will meet all requirements listed above while maintaining a visually pleasing user interface. Any conditions that have not been fulfilled will promptly notify the user to make corrections. Lastly, all visual components are easily identifiable, and their functionality are clear.

**Procedure and Actual Results:** Users will first need to choose a friend to send a drop to. This can be done in two ways, the first being navigating to the “Friends” page and tapping the friend widget and selecting “Send Drop”(Figure 3.6). This will direct the user to the “Create Drop” page where their specified
recipient is selected. Another option would be to press the floating “+” button found on any primary page (Home, Friends, Account). This will direct users to the “Create Drop” page where users can click another “+” button which will open a pop-up widget displaying their friends list (Figure 3.7).

Figure 3.6: Actions For User Jeffry.
Figure 3.7: Drop Creation Page.

Once the recipient field is no longer empty, users can then choose their desired drop locations from three options. First, users can send their drop to their recipients' preferred drop location by clicking the toggle button provided. If the toggle button is enabled all other options will be visually unavailable until the toggle is switched off. Secondly, users can provide an address they would like to send their drop to, which will update the google map to display their desired
address location. Lastly, users can choose a location anywhere on google maps and provide a custom description of their desired location. If users have already sent a drop before they can also choose a recent or favorites drop locations.

Once users are satisfied with their drop location, users can then tap the “Set Location” Button found on the bottom of the screen (Figure 3.7 Right). Users will then be directed to the Add Media page where they are required to add a message. Users can also add Media in the form of a picture or a 10s video that can be taken in the moment or selected from their device’s storage. In Figure 3.8, it can be seen how the user interface adapts to adding the media content to the drop. Users have the option to remove the media by tapping the “X” next to the content container. It should be noted that if users do not provide a message, the Udrop's client will alert them to provide one.
Figure 3.8 Drop Media Page After Providing A Custom Location.

Finally, once the user is satisfied with their drop they can press “Send Drop” where depending on the size of their content, they will see a loading screen. The loading screen indicated that Udrop is still uploading content to Udrop’s backend. Once this is completed users will be taken to the home page where they can see their newly created drop (Figure 3.9).
Now that the drop has been sent, the receiver will be notified immediately and populate their notification page with a new notification. The receiver can then navigate to their notification page via a local phone notification (Figure 3.10 Left) or view the drop from the home page (Figure 3.10 Right). Using either method, the receiver will be able to navigate to the drop location.
page (Figure 3.11 Left) where they will be able to view their location and the drop location. Once the user is within the 30-meter range of the drop, indicated by the drop zone graphical component, the receiver will be automatically directed back to the home page with an unlocked drop (Figure 3.11 Right). Both users, the sender and receiver, will be able to provide feedback on the drop using the corresponding comments page (Figure 3.12)

Figure 3.10 Notification And Home (Locked) Page.
Figure 3.11 Drop Location And Home (Unlocked) Page
Figure 3.12 Comments Page.
REFERENCES

“Mongodb Realm.” *MongoDB*,

“React Native · Learn Once, Write Anywhere.” *React Native RSS*,
https://reactnative.dev/.