**DropPoint**

*Design Specification*

Glenda Garcia & Brenda Garcia

Capstone Project

November 22nd, 2022

**Table of Contents**

[Project Overview 3](#_Toc120041950)

[Main Goal 3](#_Toc120041951)

[Budget, Time Constraints and Target Audience 3](#_Toc120041952)

[Aesthetics 3](#_Toc120041953)

[Functionalities And Features 4](#_Toc120041954)

[Performance (where the product will be used) 5](#_Toc120041955)

[Non-Functional Details 5](#_Toc120041956)

[Database Diagram 6](#_Toc120041957)

[High-Level Architecture of DropPoint 7](#_Toc120041958)

[Class and Object Design/Modules 7](#_Toc120041959)

[Sign Up Activity 7](#_Toc120041960)

[Login Activity 8](#_Toc120041961)

[Navigation Activity 9](#_Toc120041962)

[Search Fragment 10](#_Toc120041963)

[Maps Fragment 10](#_Toc120041964)

# Project Overview

The intent of the project is to plan, design, and create a mobile application that enhances user experience along the process of planning and managing a trip. It can be a tedious process when looking for locations to visit, resulting in stress, and time lost researching various Internet sites for recommendations. With the *DropPoint* mobile application, users will have one platform with all the essentials needed for optimizing the search for places to visit, advice from travelers or natives about the country, and the best places to take pictures at.

# Main Goal

As travel fanatics, we understand the struggle, and hardships that come from planning a trip, therefore, *DropPoint’s* main goal is to address this issue, and increase user experience when planning to travel.

# Budget, Time Constraints and Target Audience

For this project, free resources will be used to develop the application and no budget is necessary for its development. The project must begin at the beginning of the 2022 Fall Semester and finish no later the end of the semester. Users simply need to sign up for *DropPoint's* free services in order to begin taking advantage of them. The app’s primary target audience is the people who like to travel and discover new places.

# Aesthetics

*DropPoint* is the name of the application because it combines the words “pin drop” and “point”. *DropPoint* should be attractive to users. The ideal color scheme for the application would include shades of white, pink, black, and gray. These colors should also be used in the app's logo, and there should be a dominant color that draws users' eyes to the app's essential functions. The logo should be some sort of pin drop that matches the color scheme used throughout the application and indicates that it is intended to be used when making travel plans.

# Functionalities And Features

*DropPoint* should have a login and/or sign-up page. Once the user is able to access the application there should be four main functionalities/activities, which should consist of a map view, search posts, add post and the user’s profile view.

* The program must present a view with the logged-in user's profile image, email, biography, and any trip folders they may have made to save or bookmark various publications.
* When a user creates a trip, the various bookmarked publications are displayed for each trip in a grid view when the user clicks on it.
* The application must display a map which contains icons that dynamically showcase the number of posts associated with the searched area in the map.
* For each click on the icons inside the map, the application must be able to display within a grid view the publications associated with the icon/pin drop.
* The system must display a more detailed view of each individual post that is requested by the user.
* The application must store the publications that the user chooses to bookmark and make the according associations with the corresponding folder that it pertains to.
* The application must implement a search view and be able to search for publications based on a certain criterion. Publications by other users, as well as the logged in user will rendered.
* The application must implement a view to add a publication which requires a user to upload an image, and specify details, such as location, hours of operation, and recommendations.
* The application must enable users to logout of the app.

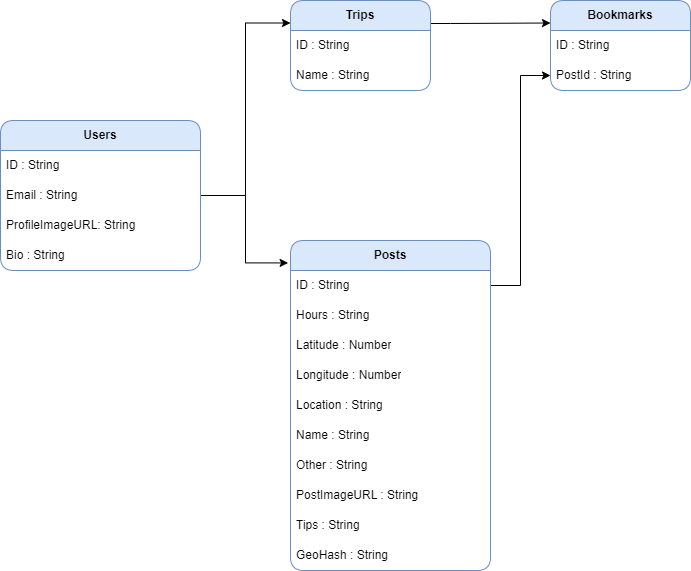
# Performance (where the product will be used)

*DropPoint* will be used by anyone who has an Android device and decides to download the application. In order to download and install *DropPoint* on an Android device, the user must access a website where the APK file for the application can be downloaded, enabling the installation to take place.

# Non-Functional Details

*DropPoint* shall implement the following non-functional requirements to the best of its capabilities given time and budget constraints:

* The user interface (UI) must adapt to the screen size of any Android device and be responsive in order to display all components without causing any to go off screen or detract from the application's aesthetics.
* In order for the password to be validated it must have at least 1 upper case, 1 lower case, 1 number, no space, and 6 to 20 characters.
* Users will be logged in automatically once they create an account without the need to enter their credentials in the login screen.
* The application should make use of a specific color schema, font, and legible size for users all throughout.
* Once a user logs in, the application should not close their session unless the user attempts to log out themselves. This implies that if a user switches to another application, they will remain logged in to *DropPoint*.
* Every unsuccessful attempt by a user when interacting with the application’s component must be recorded as a toast message, or log to keep record of errors in the system.
* The application will be accessible and will handle requests 24/7 from any place in the world that has an Internet connection.

Database Diagram****

High-Level Architecture of DropPointDiagram

Description automatically generated

# Class and Object Design/Modules

This section provides the details of the main activities and fragments used in the modules that were designed for the system.

## Sign Up Activity

Class SignUpActivity {

private editEmail, editPassword, editBio;

private imageURI;

Override void onCreate METHOD of SignUpActivity () {

Initialize connection with Firebase authorization, database, and storage

}

METHOD void createAccount () {

email = editEmail.getText()

password = editPassword.getText()

bio = editBio.getText()

isAllFieldsChecked = CheckAllFields();

if (isAllFieldsChecked)

storeUserData(email, password, bio);

}

METHOD boolean CheckAllFields () {

Check that none of the fields are empty (validation)

if (imageURI is empty)

return false

if (editEmail is empty)

return false

if (editPassword is empty)

return false

if (imageURI is empty)

return false

return true

}}

## Login Activity

Class LoginActivity {

private editEmail, editPassword;

Override void onCreate METHOD of LoginActivity() {

Initialize connection with Firebase authorization

}

METHOD void userLogin () {

email = editEmail.getText()

password = editPassword.getText()

isAllFieldsChecked = CheckAllFields();

if (isAllFieldsChecked)

if (login is Successful) {

Redirect user to Navigation View

Else

Throw error message

}}

## Navigation Activity

Class NavigationActivity {

private editEmail, editPassword;

Override void onCreate METHOD of NavigationActivity() {

switch (navigation menu item clicked) {

case map:

replaceFragment(new MapsFragment());

case search:

replaceFragment(new SearchFragment());

case add post:

replaceFragment(new AddPostFragment());

case profile:

replaceFragment(new ProfileFragment());

}

}}

## Search Fragment

Class SearchFragment {

Override void onCreate METHOD of SearchFragment() {

Initialize connection with Firebase database

Initialize connection with Places API from Google

searchResultOfLocation(); //Load result of search

}

METHOD void searchResultOfLocation () {

Populare array with cities found within 30km of the coordinates of the location selected based on latitude and longitude

}}

## Maps Fragment

Class MapsFragment {

Override void METHOD onMapRady() {

Position the map based on coordinates

Request access to user’s location

Fetch posts from database

Initialize Google’s Cluster Manager and set with posts

On click of pin drop, replace with MapPostsFragment to show posts found within selected location

}}