DiveBox

GitHub Username: perklun

Description

Wherever you go, DiveBox makes it easy for you to log your dives! With DiveBox, you don't have to worry your dive log anymore on your next diving adventure as all your dive history will be available on your mobile phone. DiveBox comes with additional functionalities such as a Twitter feed of the latest news in scuba and online storage of your log and diving photographs.

Intended User

The app is intended to be used for anyone who scuba dives.

Features

DiveBox allows users to log their dive sites by:

- Creating a profile page through their Google account
- Providing a form to log their dives
- Using Google Maps to capture the location of the user when creating a dive
- Allowing users to upload pictures for each individual dive
- Storing all dive details online in Firebase
- Providing additional storage of photos via an online storage provider such as Box

DiveBox users can view their dive history by:

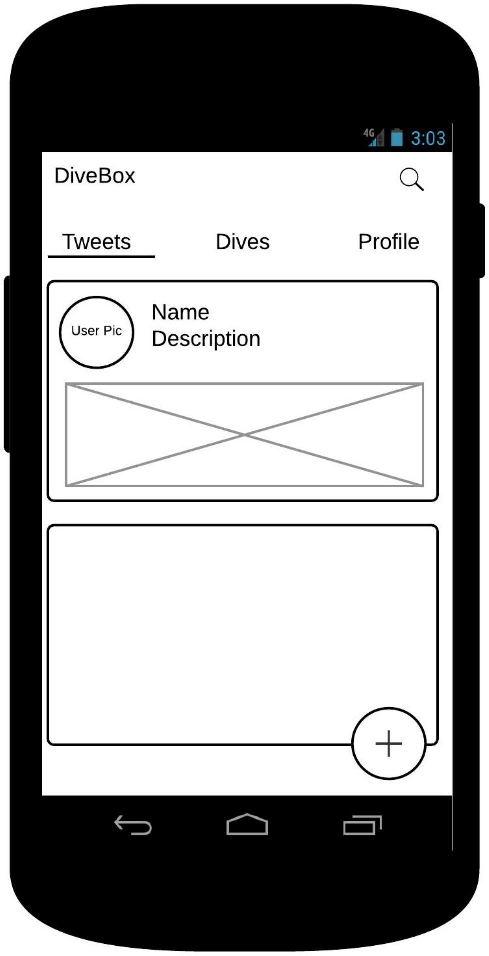
- Viewing their dive log either in map or list view
- Going into detailed view of their dives
- Previewing the pictures they took for each dive

DiveBox users provides the latest Twitter feeds in diving by:

- Connecting to Twitter via API
- Retrieving the latest feeds on scuba diving from PADI, CMAS, NatGeo, etc

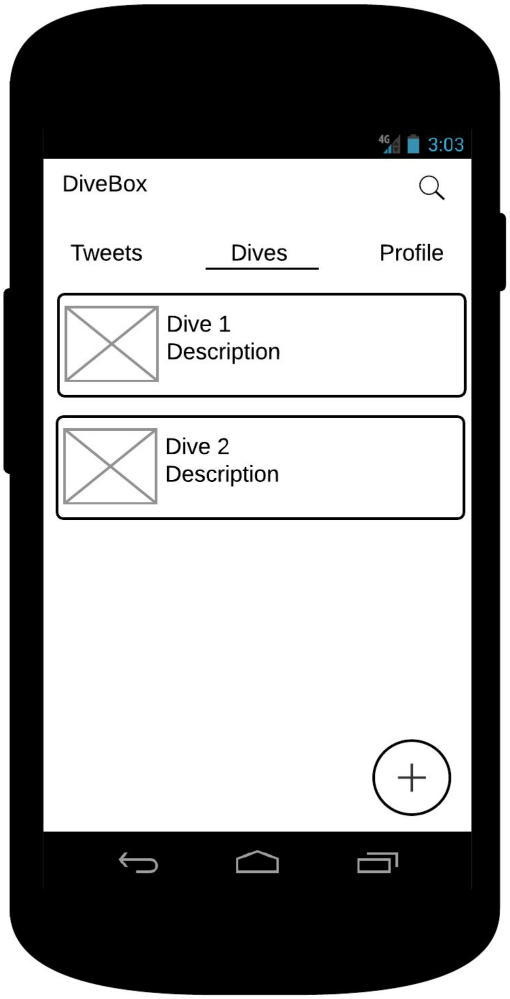
User Interface Mocks

Dive Feed



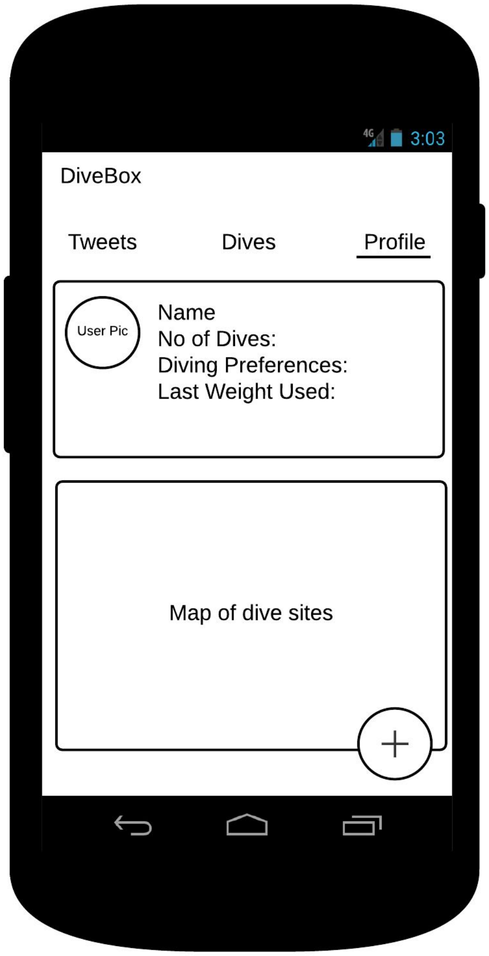
This screen shows the various Twitter feeds from diving organizations in a cardview format. The user can scroll down the cards to view more tweets until there are no more tweets to display. The tweets are ordered according to date, with the most recent at the top. By tapping on each card, more details about the tweet can be obtained by opening up their Twitter app. The floating action button brings the user to log a new dive.

Dives Page



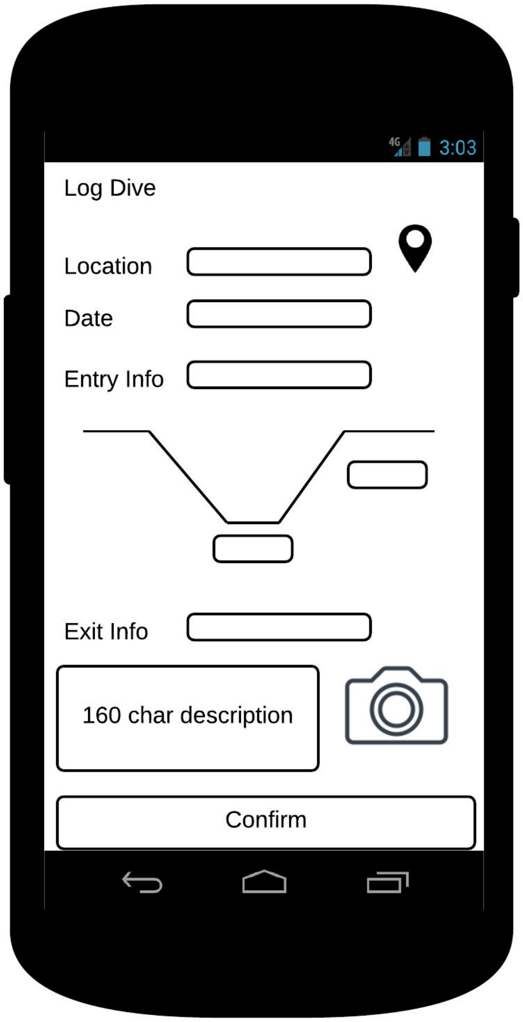
This screen displays the dives that the user has logged, sorted according to more recently updated. The search bar allows the user to find a dive easily. By tapping on an item in the list, the user is taken to the detailed view of the dive.

Profile Page



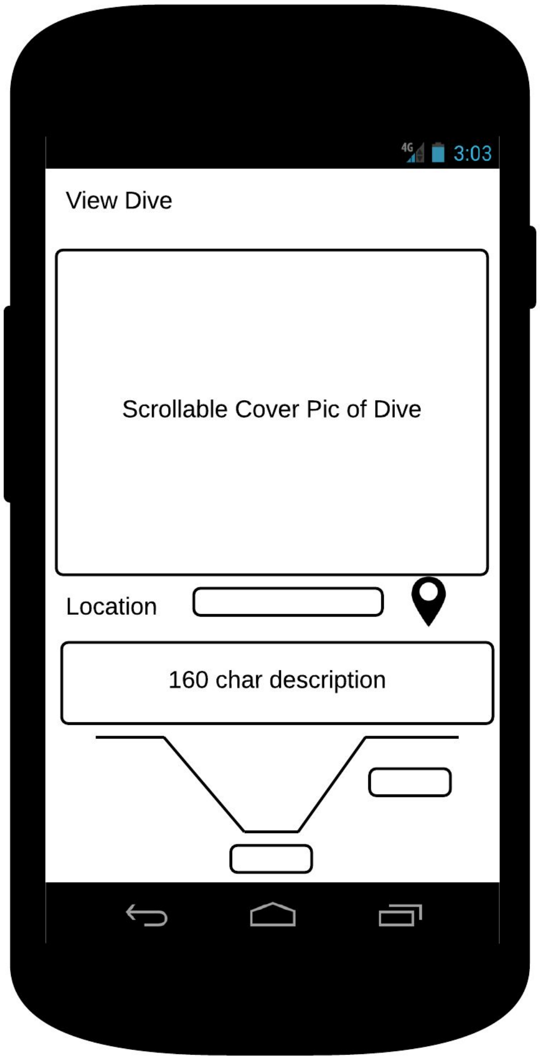
This is the profile view of a user. It contains a picture and some information about the user. This information is captured from the Google profile of the user or during the first time the user creates an account in DiveBox. The bottom half of the screen shows a map view of all the dive sites the user has created. The floating action button brings the user to log a new dive.

Create Dive



An user can log a dive with this page. It captures details about the dive from the user's input. The user can also use the Google Maps feature to determine the location of the dive when the user is at the site. After the user presses confirm, all the data will be both stored locally and on an external web storage. Photos can be added for the dive too, and also transferred to an external storage server for persistence.

Dive Details



This page displays all the information about a dive when the user taps on a dive item either in the dive page or in the profile page. The top of the page shows pictures of a dive and allows the user to scroll through them from left to right. By pressing on the location button in the middle, the user will be brought an expanded view of the location in Google Maps.

Key Considerations

How will your app handle data persistence?

The user details and dive logs will be stored locally in a SQL database. Images will be stored separately on the disk. In order to ensure that when a user re-installs an app, the information is still present, data is also uploaded to Firebase and images stored in Box for persistence purposes.

Describe any corner cases in the UX.

If users decide to cancel the creation of a dive, they will need to press the cancel button on the create dive page. However, if they want to place the creation on-hold, they should press the back button at the top left or the back button on the phone and all the data that is being entered will be saved temporarily in memory. Once the user returns to creating a dive, the information will re-populate.

When a user is done viewing a dive, the user can either press the back button at the top left or the back button on the phone to return to the previous activity.

Describe any libraries you'll be using and share your reasoning for including them.

Image loading: Picasso - easy to format and load pictures into view
API requests: Retrofit - removes the need to custom write HTTP requests
Maps: Google Play Services - provide mapping for location determination

Next Steps: Required Tasks

Task 1: Project Setup

- Create project in GitHub
- Setup Twitter API Key
- Setup FireBase account
- Setup Box account (optional)
- Obtain Google Play Services permissions

Task 2: Implement UI for Each Activity and Fragment

- Create 3 activities:
    - Login Activity
    - Tweets/Dives/Profile Activity
    - Create Dive Activity
    - View Dive Activity
- Create ViewPager for Tweets/Dives/Profile Activity
- Build fragments for each activity
- Build layout for each fragment

Task 3: Manage Network Requests

- Implement API request for account creation and sign in
- Implement Twitter API request to request Twitter Feeds
- Build network requests to retrieve information from Firebase

Task 4: Populating and storing data

- Build adapters to load dive logs and user data
- Implement methods to manage data persistence in SQL database
- Build methods to temporary store user data during dive log creation

Task 5: Advanced UI features

- Create scrollable view of pictures in View Dives
- Create map view of dive locations
- Create adapters to allow user to scroll infinitely through Twitter feeds

Task 6: Manage Intents and services

- Create explicit intent to launch Twitter app when tweet is tapped
- Build methods to manage import of pictures to dives
- Build service to sync data between app and online storage when dive is created offline

Task 7: Polish the App

- Create animations where appropriate
- Polish UI to ensure app runs smoothly and provides a good user experience
- Test the app for any crashes and bugs