GitHub Username: anandbibek

Parallel Share

Description

The app will be a compact and minimal social networking client that can post text(+ images) to different social networks simultaneously. Facebook and Twitter will be supported via API and Google+, which lacks an API, can also be included using on-device intents.

Intended User

Users who are frequently posting status updates and pictures to different social networks like facebook, twitter and G+, will find this app highly useful. It can streamline their experience by providing a unified interface for uploading tweets, facebook updates or pictures.

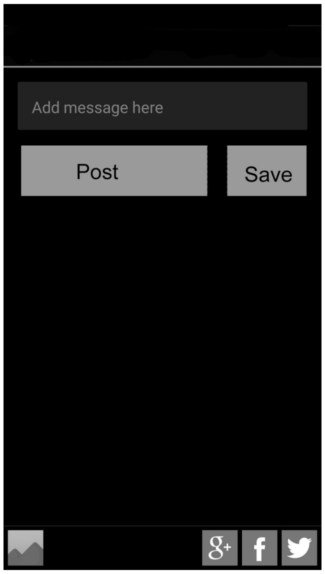
Features

- Connects to Facebook.

- Connects to Twitter.
- Authorizes and saves login information of the social networks.
- Allows user to select which networks to post, from Twitter, Facebook and Google+.
- Posts text and/or images to selected social networks simultaneously.
- Allows user to save drafts of posts that can be accessed later.

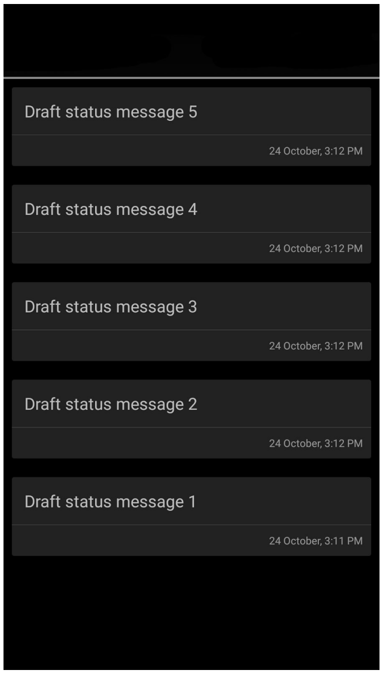
User Interface Mocks

Screen 1



Main UI for posting or saving posts and status updates. The buttons below will provide toggle to select social networks to post.

Screen 2



Draft status message 5

24 October, 3:12 PM

Draft status message 4

24 October, 3:12 PM

Draft status message 3

24 October, 3:12 PM

Draft status message 2

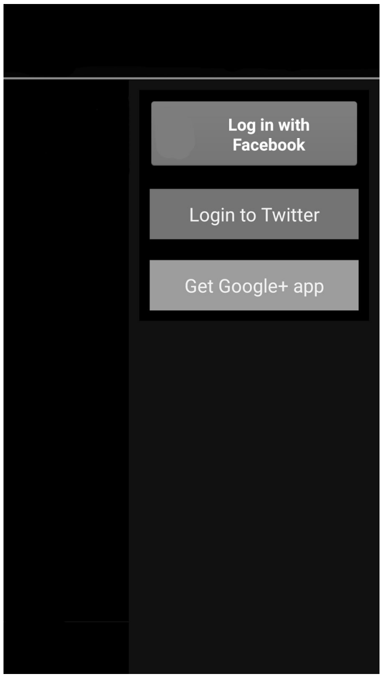
24 October, 3:12 PM

Draft status message 1

24 October, 3:11 PM

Screen where saved status updates can be seen and clicked to use.

Screen 3



Screen with login options for various networks.

Key Considerations

How will your app handle data persistence?

User account information will be stored in SharedPreferences. Posts will be stored in SQLite database and wrapped by content provider.

Describe any corner cases in the UX.

The API calls to post to social networks will have to be handled properly so that it can propagate independently of the UI, but it has to provide a mechanism to cancel in case it is requested by user.

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

Use Google Play Services Admob to display test ads.
Use Google Play Services Analytics to analyse user behavior.
Use Android support library to design UI.

Describe how you will implement Google Play Services.

Admob: By adding in-app ads to generate revenue.
Analytics: By getting users' behavior data in using the app (time spent in certain pages, usage frequency of certain features/buttons etc.) to analyse user preference and make improvement.

Next Steps: Required Tasks

Task 1: Project Setup

- Create base Android project in Android Studio.
- Setup gradle dependencies, libraries.
- Create client based API access tokens for Twitter and Facebook.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for post writer fragment
- Build UI for drafts display fragment

Task 3: Define interface for handlers and uri for content provider

- Define interface in all handlers which will be used by UI
- Define uri scheme for content provider

Task 4: Build SQLite Database

- Create database to store user drafts and posts information
- Implement content provider to access stored data

Task 5: Authorization

- Create the flow to authorize user via oAuth via Facebook and Twitter API.
- Store the auth tokens for future use and update when they expire.

Task 6: Handle Content posting

- Create functions to post text and images to Twitter and Facebook through proper API calls.
- Show progress indicator and results of pos action.
- Create function to store data in database for drafts section.

Task 7: Google Play Services

- Implement in-app ads.
- Implement play services : analytics.

Task 8: Widgets

- Add a widget for quick access from home screen.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”