

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[1. Home \(MainActivity\)](#)

[2. Side Menu](#)

[3. MeetupMainView](#)

[4. GithubMainView](#)

[5. Search Dialog \(Fragment\)](#)

[6. MeetupGroupDetailView](#)

[7. MeetupEventDetail](#)

[8. GithubDetailView](#)

[9. Bookmarks](#)

[10. Widget](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup, MainActivity, Side Menu](#)

[Task 2: MeetupMainView, GithubMainView, Search Dialog](#)

[Task 3: MeetupDetailView, BookmarkedEvents, Widget](#)

[Task 4: GithubDetailView](#)

[Task 5: Settings, Notifications](#)

[Task 6: Polish UI](#)

GitHub Username: ootahiaoo

Daily Update

Description

An application to help developers stay updated with their topics of interest. You can quickly know if you are missing any recent hot repository or nearby event related to your interests.

Intended User

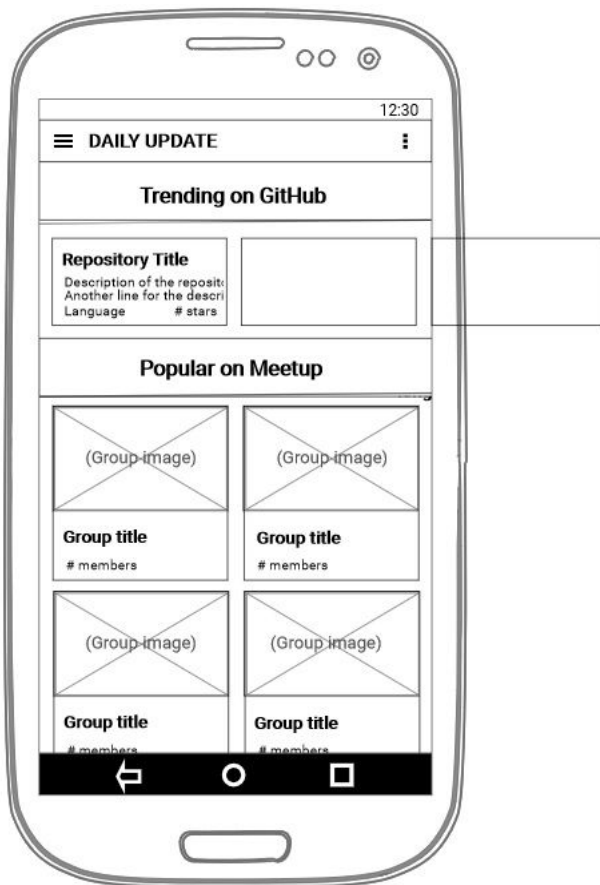
This application is intended for developers who like to discover new repositories on GitHub and participate to Meetup events.

Features

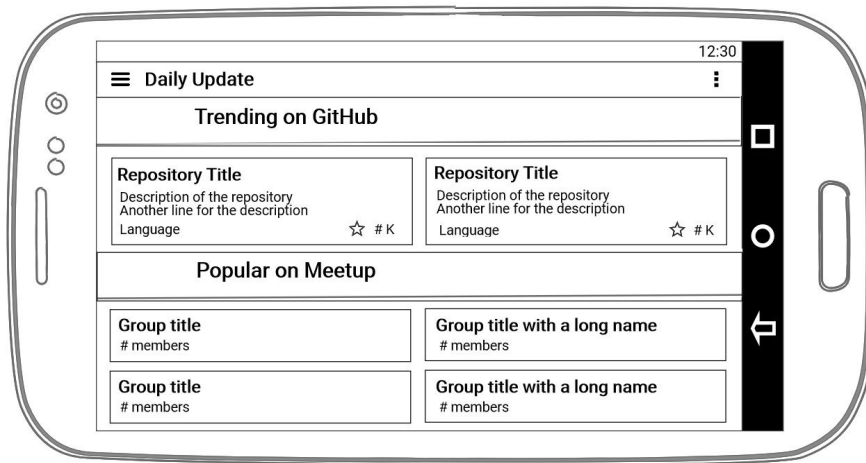
- Retrieve information from several sources (Github, Meetup).
- Set up a search based on keywords and display with preferred sort order.
- View the details of a repository or an event within the app or through the user's preferred browser.
- Bookmark an event to view details later.
- Get notifications as soon as there is any new event matching your search.
- Have a widget display bookmarked events.

User Interface Mocks

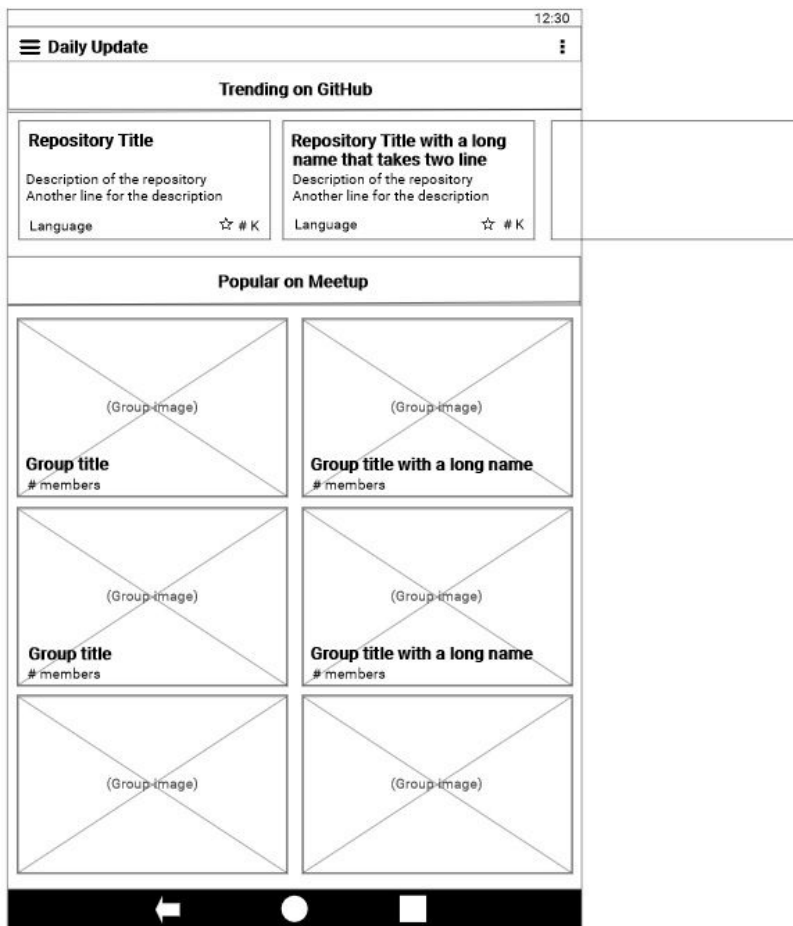
1. Home (MainActivity)



(Mobile portrait)



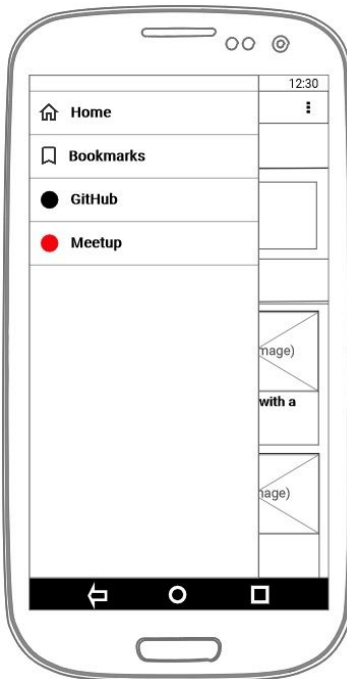
(Mobile landscape)



(Tablet portrait)

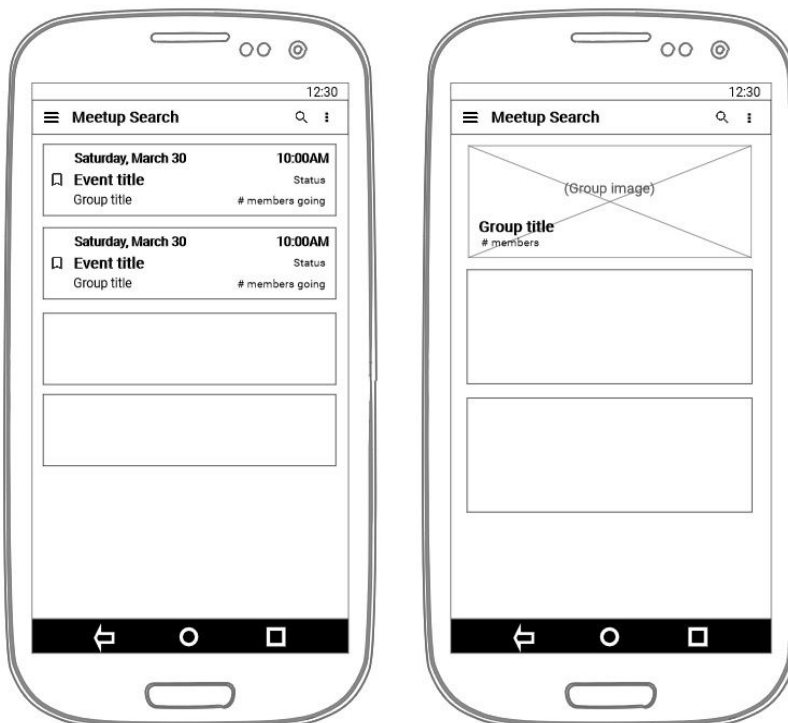
This is the first screen that appears when the user opens the app. It displays the trending repositories on GitHub and the popular Meetup groups in the Tech category (the groups will be limited to those near the user location if the user gave the permission to use the device location).

2. Side Menu

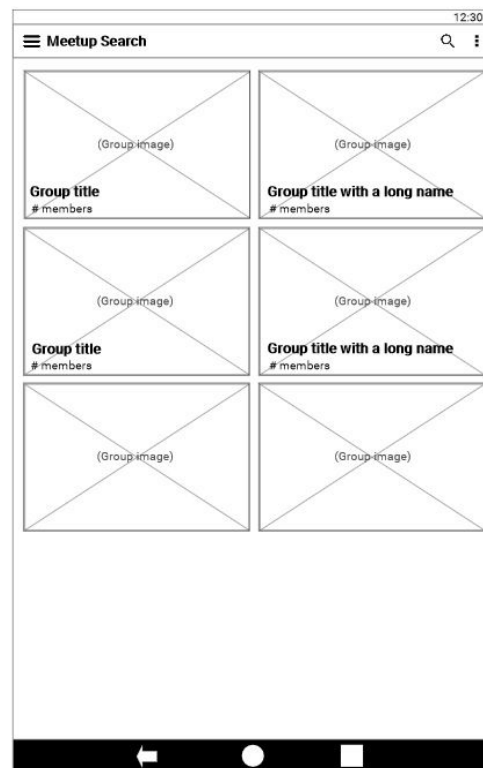


This is the menu that is displayed when the user hits the "hamburger icon" of the action bar on the top left. The menu allows the user to navigate between the user's bookmarked events (Bookmarks), GitHub repository search view (GithubMainView) and the Meetup event/group search view (MeetupMainView).

3. MeetupMainView



(Mobile portrait: sort by calendar & sort by groups)

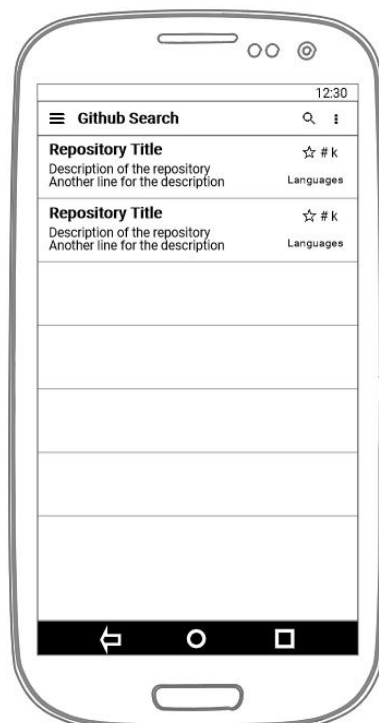


(Tablet portrait: sort by calendar & sort by groups)

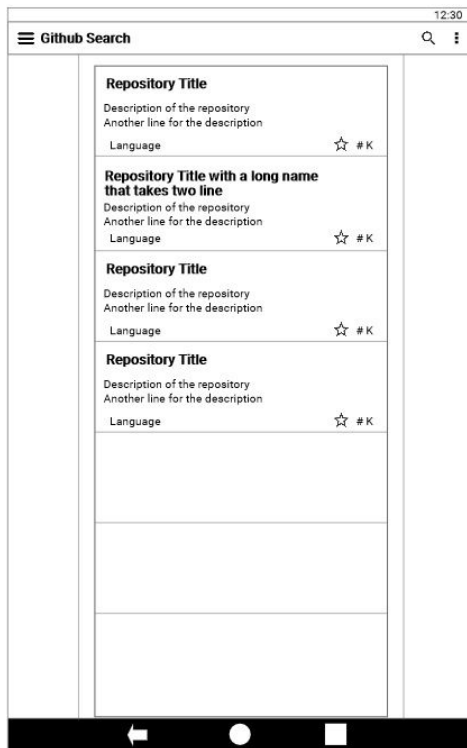
The user can search for Meetup events or groups from this view. If the user has already used the app, it will display the search result using the criterias saved from the previous search.

The user can bookmark/unbookmark the events from the list.

4. GithubMainView



(Mobile portrait)



(Tablet portrait)

The user can search for GitHub repositories from this view. If the user has already used the app, it will display the search result using the criterias saved from the previous search.

5. Search Dialog (Fragment)



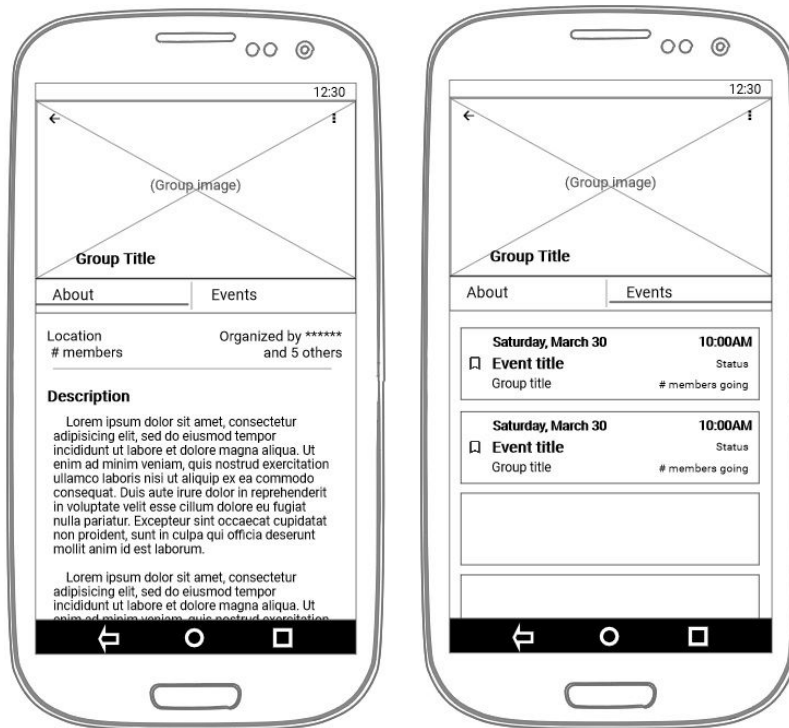
(Mobile portrait)

REPOSITORY SEARCH OPTIONS	
Keywords:	<input type="text" value="Input keywords"/>
Language:	<input type="text" value="Java, PHP, Python..."/>
Sort by:	<input checked="" type="radio"/> Updated <input type="radio"/> Stars <input type="radio"/> Forks
Order:	<input checked="" type="radio"/> Descending <input type="radio"/> Ascending
<div>Cancel SAVE</div>	

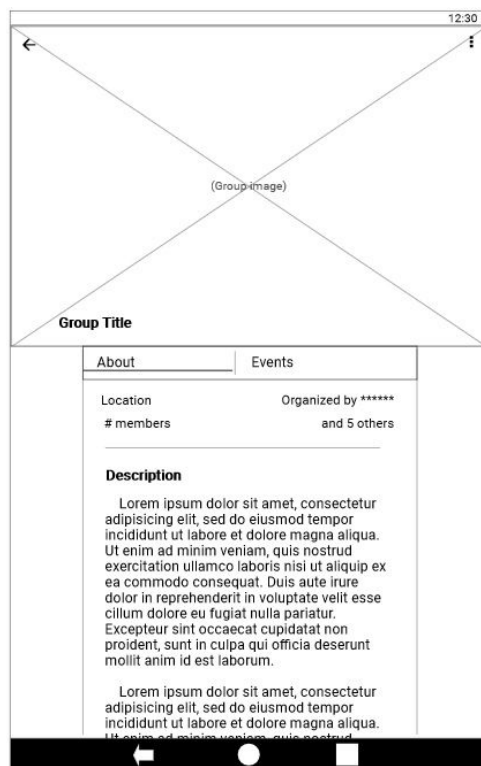
This is the dialog displayed in GithubMainView and MeetupMainView when the user hits the "search" icon of the action bar.

The user can input or edit search criterias from this dialog. The dialog is similar for both views, but the search options are different.

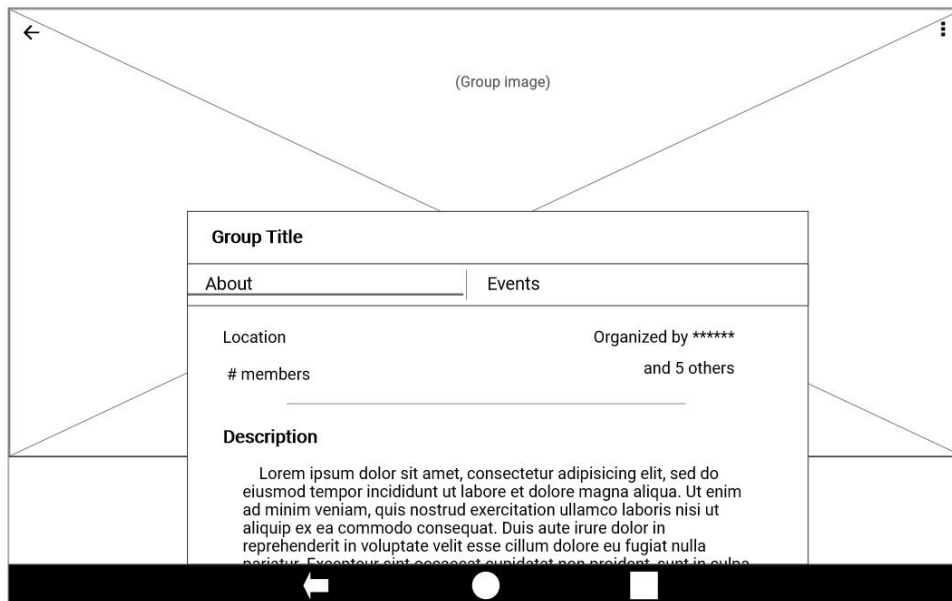
6. MeetupGroupDetailView



(Mobile portrait)



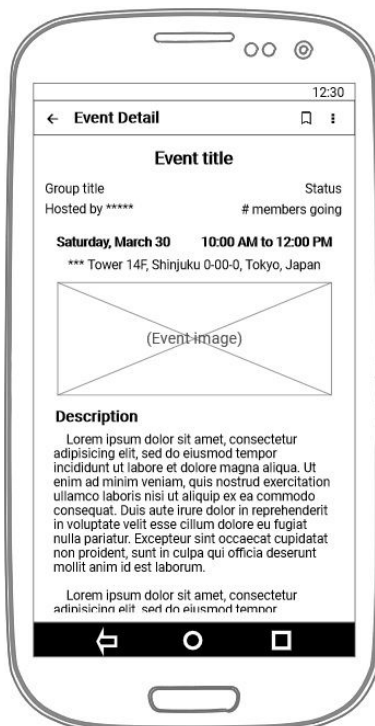
(Tablet portrait)



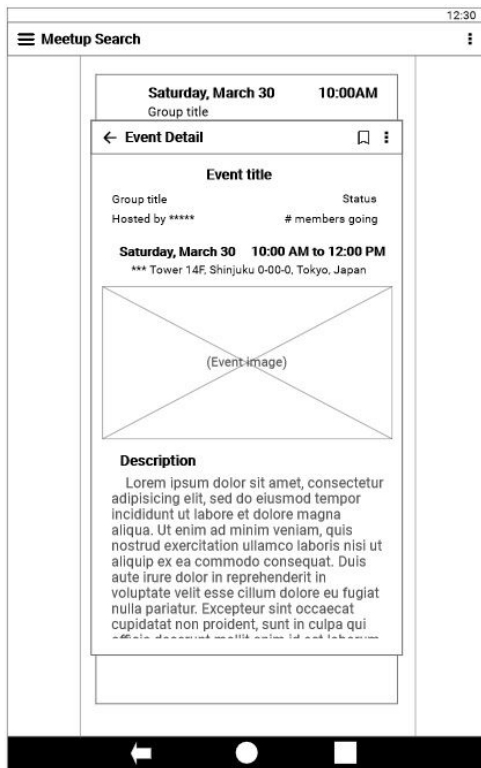
(Tablet landscape)

This is where the user can see the details of a Meetup group. The group description and events related to that group are separated in two tabs. The user can bookmark/unbookmark events from the events list.

7. MeetupEventDetail



(Mobile portrait)

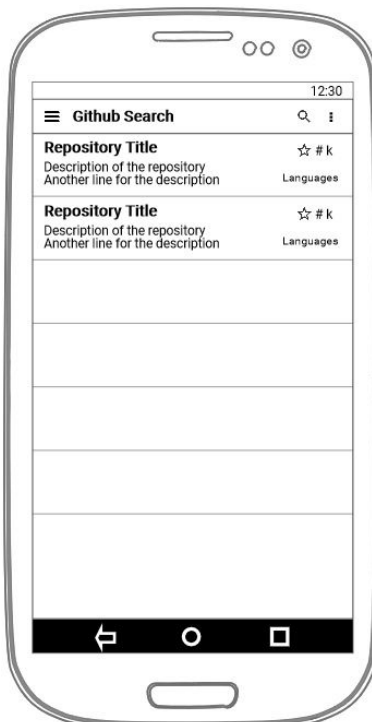


(Tablet portrait)

The details of a Meetup events are displayed in a fragment. The user can bookmark/unbookmark the event from this view.

The fragment is displayed on top of the search result or group detail view in tablet mode.

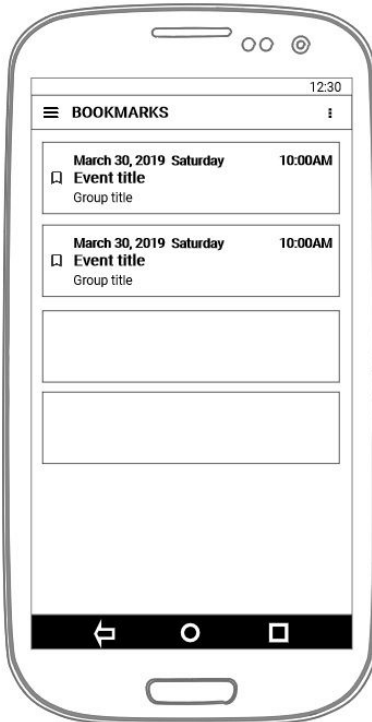
8. GithubDetailView



(Mobile portrait)

Selecting a repository item from the search list (in GithubMainView) will open the repository details in its web-version through Chrome Custom Tabs.

9. Bookmarks

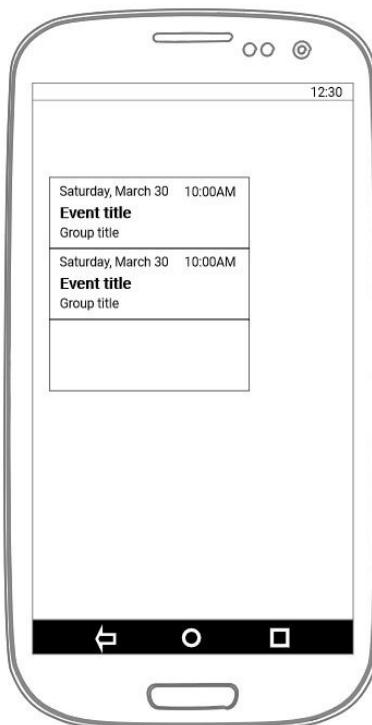


(Mobile portrait)

This is where the bookmarked events are displayed. It only includes the date, time, event title and group title, in order to let the user see this list offline.

Selecting an item will open the MeetupEventDetail view, where the latest information will be fetched from the Meetup API.

10. Widget



The app widget displays the bookmarked events list.

Key Considerations

How will your app handle data persistence?

- LiveData to keep the current search result open if the app is put in background (for example, when viewing repository details).
- SavedInstanceState to temporarily save the user input in case the app is put in background while inputting search keywords.
- SharedPreferences to save the search criterias.
- Room to save a bookmarked event information.

Describe any edge or corner cases in the UX.

- When the user taps on a repository in the GithubMainView, instead of making an in-app design for the repository details, the app will use Chrome Custom Tabs to display the web-version of the repository. A custom menu will allow the user to open the page in their favorite browser. Hitting the back button will bring the user back to the search list (GithubMainView).
- Pressing the back button on MeetupMainView or GithubMainView will bring the user back to the Home (MainActivity).
- The event details are displayed in a fragment that is displayed on top of the MeetupMainView or the MeetupGroupDetailView in Tablet version. The "side menu" icon is replaced by a "back" icon for this view.

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

- Retrofit for network calls (cleaner code).
- Glide to load/display images (improve speed and caching).
- Butterknife to bind the views (less boilerplate code).
- Chrome Custom Tabs to display GitHub repository details (the user is less confused with the web-version of GitHub and can navigate through the code more easily).

Describe how you will implement Google Play Services or other external services.

- Google Location Service: The location permission will be asked when the user first opens the app, then it will be used to search for events nearby. In case the user refuses to give permission, then they will have to input a location manually for each search.
- FirebaseJobDispatcher: The JobDispatcher will pull data from the API(s) to know if there is any new event added on Meetup matching the user's search. If there is, then a notification will warn the user about it. The notifications can be turned off in the settings.

Next Steps: Required Tasks

App is written solely in Java and uses the latest stable releases for Android Studio, Gradle and all libraries.

Task 1: Project Setup, MainActivity, Side Menu

- Create project on Android Studio 3.3.2 using Gradle 5.3.1.
- Build UI for MainActivity (Home).
- Build UI for the side menu (DrawerLayout).
- Configure Glide v4 (4.9.0), Butterknife 10.1.0 to prepare to receive data.
- Configure GoogleLocationService 16.0.0 (implementing 'com.google.android.gms:play-services-places:9.8.0' and 'com.google.android.gms:play-services-location:9.8.0') to get the user location.
- Configure Retrofit 2.5.0 to retrieve information from Github and Meetup APIs.
- Make the RecyclerViews display popular meetups (based on the user location) and trending repositories.

Task 2: MeetupMainView, GithubMainView, Search Dialog

- Build UI for MeetupMainView (RecyclerView, Paging, Toolbar).
- Build UI for GithubMainView.
- Build UI for the search dialog (Fragment).
- Implement search logic based on user input.
- Make the MeetupMainView and GithubMainView display the user searches.
- Implement SharedPreferences to save the user's searches.
- Implement LiveData to temporarily save the user's current search result (being currently displayed) in case the app is put in background.
- Make sure SavedInstanceState save the user input in case the app is put in background while inputting search keywords.
- Create a dialog that will appear when the user long-click on a paging item title. The dialog will allow the user to edit the search keywords or delete the search.

Task 3: MeetupDetailView, BookmarkedEvents, Widget

- Build UI for MeetupDetailView
- Build UI for BookmarkedEvents
- Implement database (Room) and LiveData/ModelView for the bookmarked events.
- Create the Widget.

Task 4: GithubDetailView

- Build UI for GithubDetailView.
- Implement Chrome Custom Tabs 23.3.0.

Task 5: Settings, Notifications

- Build UI for the Settings (Fragment).
- Implement Settings options:
 - "Share (event/repository)" option to share the currently viewed event or repository.
 - "Refresh" option.
 - "Copy Link" and "Open in (user's preferred browser)" options (only available for GithubDetailView)
 - "Reset search" option to delete all the saved search criterias.
- Configure FirebaseJobDispatcher 0.8.6 to get notifications for new events, then include option to turn the notifications off in the settings.

Task 6: Polish UI

- Create tablet layouts for Home, MeetupMainView, GithubMainView, GithubMainView, MeetupGroupDetailView.
- Add the logic to put the MeetupEventDetail fragment appear in front of the MeetupMainView and MeetupGroupDetailView when using a tablet.
- Create transitions.
- Final check:
 - Make sure all inputs from servers and users are validated.
 - Add comments on methods where necessary/better to have.
 - Put all strings are in a strings.xml file to facilitate future translation.
 - Accessibility support related: follows Material Design guidelines, includes content descriptions for images/icons, all dimensions are defined in dp and texts in sp, enables RTL layout.