GitHub Username: amha

Days Off

Description

Coordinating a group event usually involves dozens of notifications, text messages, phone calls, and emails, which is exhausting. Days Off (or D.O.) is a tool that makes it easier to coordinate free time with friends and family.

To do so a user simply selects the day's they're off, time of day, and the type of activity they'd like to do. From there, the user has the ability to their calendar with their contacts.

Intended User

The target audience are men and women between the ages of 30-49 who are 1) frustrated by the process of setting up group events, 2) take the initiative organizing groups of people, and 3) less engaged with social media platforms.

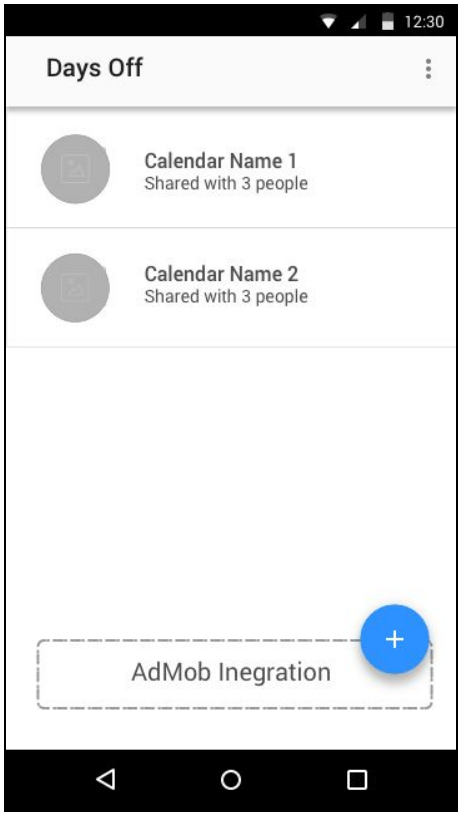
Features:

- Tracks of Days Off
- Share a calendar with contacts
- Share days off with friends via SMS

User Interface Mocks

MainActivity

The main activity is a list view of Calendars. Tapping on a list item will launch the CalendarDetailActivity.

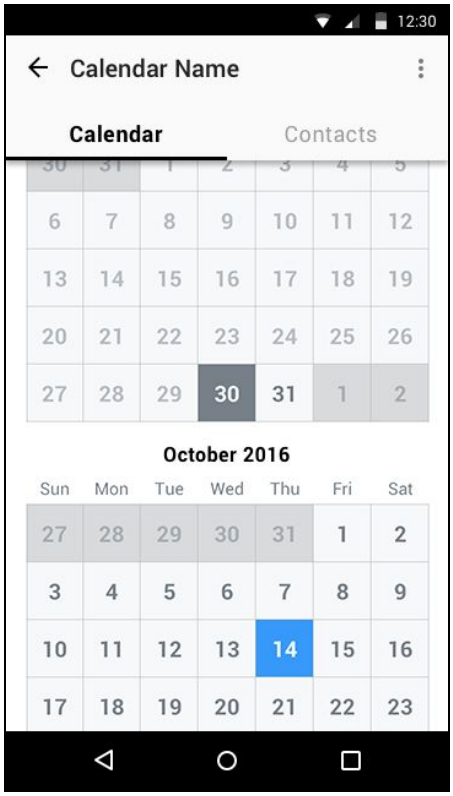


.

CalendarDetailActivity

This screen has 2 tabs, where the user can toggle between a calendar view and a contact list. Users will be able to control access to the calendar through options in the overflow menu.

.



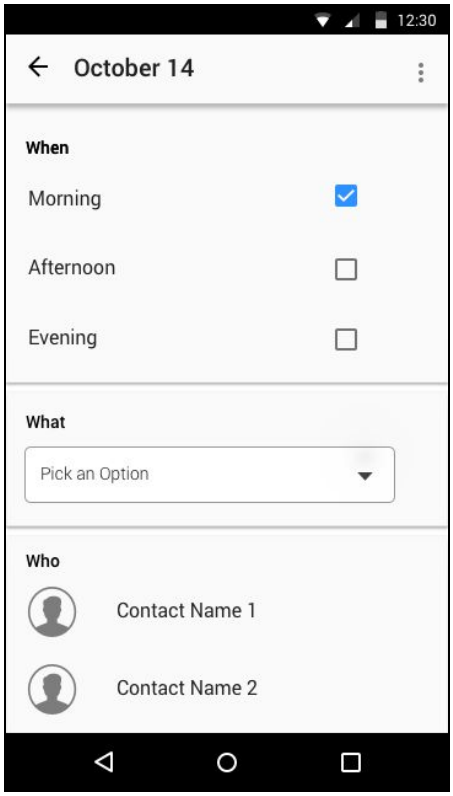
The contacts tab:

.

EventDetailActivity

The event detail screen allows the user to quickly select their availability (morning, afternoon, or event) and what they would like to do (grab food, watch a movie, etc…)

.



Key Considerations

How will your app handle data persistence?

The app intends to use tables that are managed by Android's CalendarProvider class. In order to implement the share functionality, the app will use the Google Calendar API as the backend.

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

- The **Google Calendar API** will serve as a backend to the application.
- Square's **Android Time Square library** will be used to drive the date selection experience.
- The **ButterKnife** library will be used to optimize data binding.

- Square's **Retrofit** library will be used to communicate with the Google Ccalendar backend.
- Lastly, the following content providers will be used to streamline I/O: **ContactProvider** and **CalendarProvider.**

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

In terms of initial setup, I'll create a new project that will support Android 4.4+ on Phones & Tablets. Next, I'll import all of the previously listed libraries through Gradle. In terms of build variants, the current development effort will support a single build variant, which will include ads. [1]

Task 2: Implement UI for Each Activity and Fragment

Version 1.0 of Days Off will support mobile and tablet devices. The screens for each of these devices will be developed in Task 2.

The key subtasks include:

- Build UI for MainActivity
- Build UI for CalendarDetailActivity and CalendarDetailFragment
- Build UI for EventDetailActivity and EventDetailActivity
- Build UI for NewCalendarAcivity and NewCalendarFragment

Task 3: Implement Local Calendar I/O

Task 3 is mainly concerned with implementing local CRUD functionality, which will leverage Android's CalendarProvider class.

The key subtasks include:

- Import CalendarProvider (and any relevant classes)
- Grant necessary system permissions for Calendar I/O
- Read/Write from existing Calendar Tables

[1] Future development efforts will depend on validation received through user research.

- Create/Delete new Calendar tables

Task 4: Integrate Google Calendar API

Task 4 extends CRUD functionality to the Google Calendar API.

The key subtasks include:

- Import necessary classes to interaction with Google Calendar
- Import Retrofit to handle network communications
- Read data from Google Calendar
- Write data to Google Calendar

Task 5: Implement Share functionality

There are 2 types of sharing: granting permission to a calendar and sending data via SMS (and or other casual communication methods).

The key subtasks include:

- Import necessary libraries from Android's ContactProvider class
- Grant necessary system permissions to read Contact from Address Book
- Implement Calendar share with Contact from Address Book
- Implement sharing through SMS, Whatsapp, etc...