GitHub Username: liewjuntung

Travel Companion

Description

Make your trip awesome by using Travel Companion! Travel Companion is your go-to apps when you want a personal assistant to help you to plan your trips.

Intended User

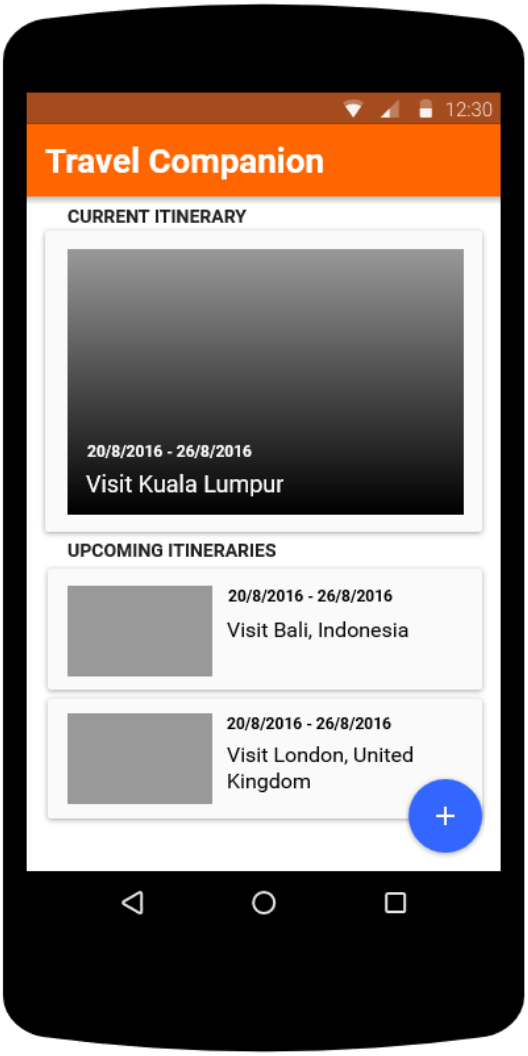
Travelers

Features

- Add new Itinerary
- Modify itinerary
- Offline capability
- Weather forecast

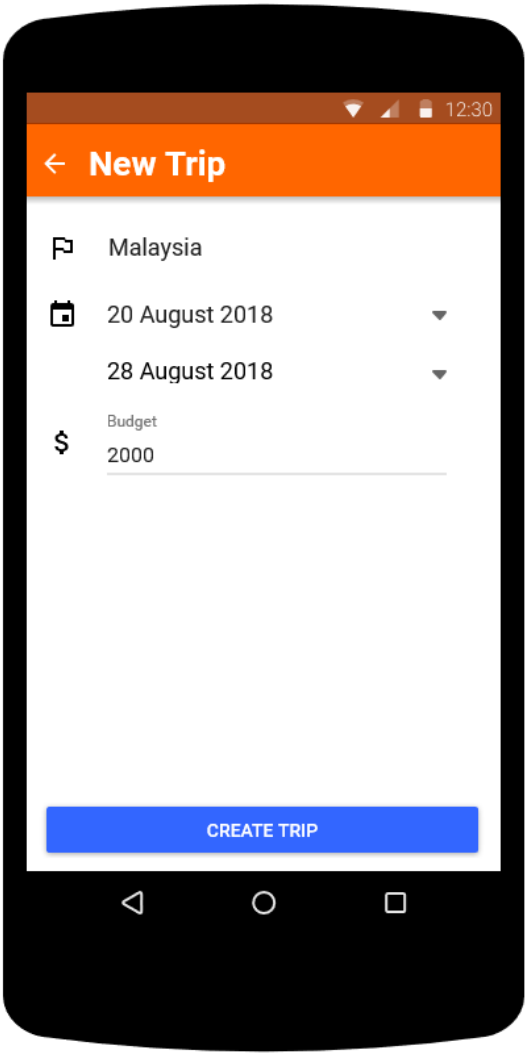
User Interface Mocks

Screen 1



The **MainActivity**, where it will display itineraries created by the users that is happening or will happen in the future.

Screen 2

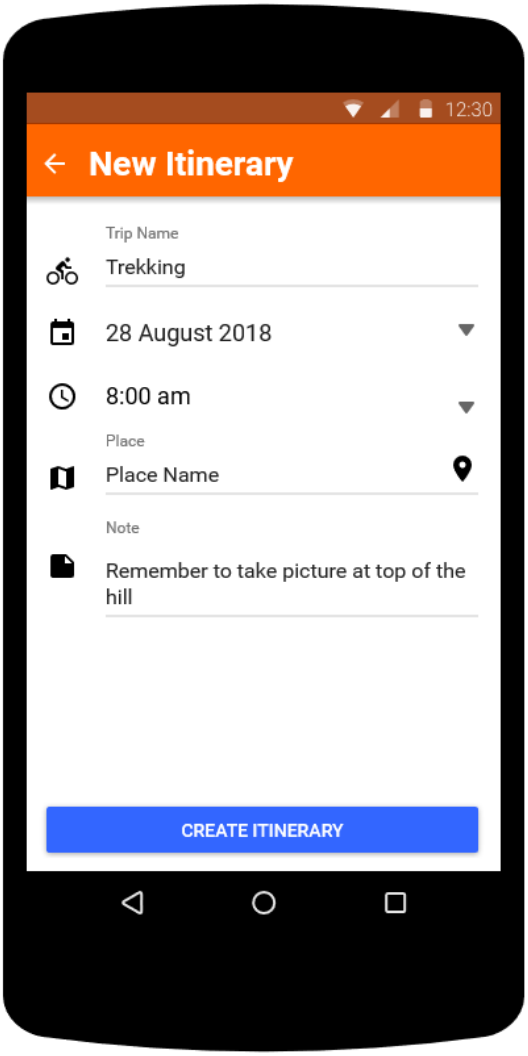


After the user tapped on the FAB button in Screen 1 it will launch **CreateTripActivity**. And the users can fill in details such as the country, date range and budget to be used in this application. After the user completed this step by clicking on the **“Create Trip”** Button, it will launch Screen 3

Screen 3

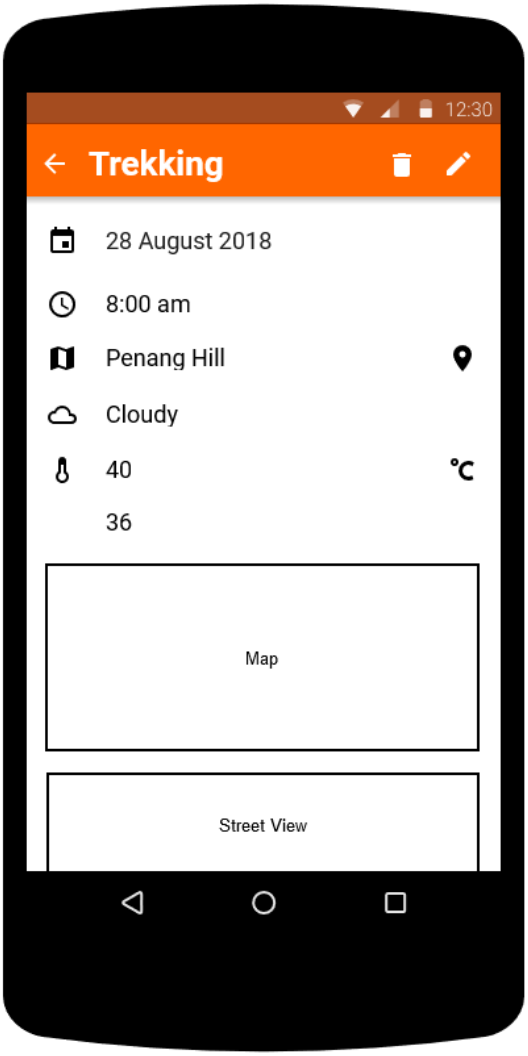
This screen is the **TripActivity** where the itinerary details are listed out to the users. It will display an image in the CollapsedToolbar layout. There will be a RecyclerView that will display current activity and upcoming activities. The items will be consisted of Time, Date, Activity's name, place, and weather condition icon that will display weather for that time if available. The Floating Action Button will launch the **CreateItineraryActivity** in Screen 4. Each item will launch **ItineraryActivity** in Screen 5

Screen 4



CreateItineraryActivity, The FAB in Screen 3, The date and time will be set via DatePickerDialog and TimePickerDialog respectively. The place can be done via typing or via Google Places via clicking on the places icon. **Create Itinerary** Button will lead to Screen 5

Screen 5



ItineraryActivity is launched by Screen 3 and Screen 4. It has use Google Map and Street View to show the location of the place of this activity. The edit button will launch Screen 4 with all the existing data. And the delete button is for deleting this activity, and back to Screen 3 if it is successful.

Key Considerations

How will your app handle data persistence?

It will utilize Content Providers achieve data persistence

Describe any corner cases in the UX.

1. Ads will be displayed as interstitial ads, they will be displayed after the user finished creating the itineraries and activities.
2. when the user scroll up in the main activity (Screen 1) and the itinerary detail (Screen 3), the floating action button will hide itself in order to not obscure the item in the recyclerviews.

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

1. Picasso
    - for loading images of the places
    - for loading pictures of countries in itinerary detail screen
    - compile 'com.squareup.picasso:picasso:2.5.2'
2. Retrofit
    - for loading weather data
    - for loading itineraries from the cloud
    - for loading itineraries up to the cloud
    - compile 'com.squareup.retrofit2:retrofit:2.1.0'
3. CountryPicker
    - for selecting the country to visit in the initialization of the itinerary
    - https://github.com/chathudan/CountryCodePicker
4. ThreeTenBP
    - For using Java 8 date and time library
    - compile 'org.threeten:threetenbp:1.3.2'
5. Schematic
    - To automatically generate content providers and database

Describe how you will implement Google Play Services.

1. Places
    - It's to get place details and address during creating entries for itineraries
2. Admob
    - Interstitial ads, for generating income by displaying relevant ads.

3. Maps
    - For users to know where to go or for easy navigation
    - StreetView for users to know the looks and feels of the place.

Next Steps: Required Tasks

Task 1: Project Setup

- Setup the Android project
- Configure the libraries needed for the this project.
- Add in the google-service.json in the project to enable Google Play and Firebase integration.
- Create Ads for AdMobs
- Add the necessary permissions in the Android Manifest.
- Add debug app id for Admobs

Task 2: Create Data and Content Providers

Create the data files that will drive the application and enable it to function offline

- Create Trip class and implement with parcelable interface
- Create Itinerary class and implement with parcelable interface
- Create Content Providers by using simonvt/schematic generator
- Create Retrofit Service and link up with the Yahoo Weather API
- Create a Utility class for reusable functions
- Create Content Providers for Itineraries that will have
    - id
    - name
    - country
    - date_from
    - date_to
    - budget
- Create Content Provider for Itinerary activities
    - id
    - name
    - place
    - latitude
    - longitude
    - time
    - date
    - weather

- Make sure content providers are stated in the Android Manifest

Task 3: Set up themes and create reusable layouts

Set up a couple of resource files for reusability

- Select primary, secondary and accent colors.xml in the resources folder
- Create dimens.xml to insert useful dimension values such as the keylines and distance between views
- Create toolbar layout in xml to be reused in other layouts
- Create activity_detail_menu with trash icon and edit icon for Screen 5

Task 4: Create UI and functions for MainActivity (Screen 1)

In the main activity class, it will have a recyclerview to display all the trips created by the users

- Complete MainActivity Layout
- Create a wrapper class for Trips and headers in RecyclerView
- Implement the RecyclerView and RecyclerView Adapter.
- Implement Cursor Loader to load trips

Task 5: Create UI and functions for CreateTripActivity (Screen 2)

Create a new activity call CreateTripActivity for creating new Itineraries

- Create layout
- Call DatePickerDialog when the date button is clicked
- Call TimePickerDialog when the time button is clicked
- Create a callback where Screen 1's FAB will launch CreateTripActivity when clicked
- Create a validation function in the utility class
- Use Content Resolver to save new trip into the database

Task 6: Create UI and functions for TripActivity (Screen 3)

Create a new activity call TripActivity, it is after the user has clicked on a trip in Screen 1 or has created a trip in Screen 2.

- Complete TripActivity Layout
- Create a wrapper class for activities and headers in RecyclerView
- Implement the RecyclerView and RecyclerView Adapter.
- Implement Cursor Loader to load itineraries in the selected trip
- Implement logic to fetching weather forecast for upcoming 6 days for each itineraries, if it's more than 6 days, display weather forecast is unavailable with a tiny text.

Task 7: Create UI and functions for CreateItineraryActivity (Screen 4)

Create a new activity call CreateItineraryActivity for creating new Itineraries

- Create layout
- Implement Google client's OnConnectedCallbacks and OnFailedListeners.
- Call PlacePicker when the place button is clicked and save the location detail for itinerary.
- Create a callback where Screen 3's FAB will launch CreateItineraryActivity when clicked
- Create a validation function in the utility class
- Use Content Resolver to save new itineraries into the database when Create New Itinerary Button is clicked and launch TripActivity (Screen 4)

Task 8: Create UI and functions for ItineraryActivity (Screen 5)

Create a new activity call ItineraryActivity, it is after the user has clicked on an itinerary in Screen 3 or has created a trip in Screen 4.

- Complete IteneraryActivity Layout
- Create OnItineraryClickListener for to implement in the TripActivity's RecyclerView (Screen 3)
- Implement Google client's OnConnectedCallbacks and OnFailedListeners.
- Implement Runtime Permission Request for Location and Maps
- Implement Google Maps and StreetView
- Implement logic to fetching weather forecast for upcoming 6 days for each itineraries, if it's more than 6 days, display weather forecast is unavailable with a tiny text.

Task 9: Create interstitial Ads

- Create an interstitial ads activity in between TripActivity (Screen 3), CreateItineraryActivity (Screen 4) and ItineraryActivity (Screen 5)
- Use SharedPreference to let enable the interstitial ads to be shown for every 10 actions