iTransmit

GitHub Username: https://github.com/Chihurumnanya

Description

iTransmit helps you organize your business and personal trips by recording your expenses, budget and the places you want to visit.

- Record travel expenses: Add new expenses with a press of a button, using the app or the home screen widget. Select the currency, the date, the category and add photos to an expense.
- Stop overspending: Create different budgets for each currency or category. It automatically converts it to your home country currency.
- Manage your itinerary: Never get lost again! Add places you want to visit for each trip and easily get directions to it.
- Never lost your data again: All your trip data is securely saved in the cloud.

Intended User

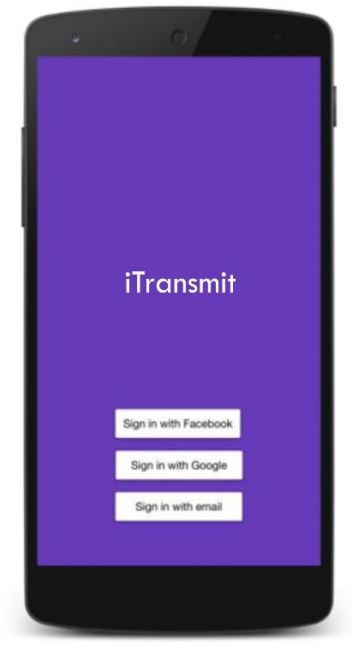
This app is targeted to users who frequently travels, are organized and have an age of18 years old and above.

Features

- Store travel expenses with currency, image and location data
- Create individual budgets for each currency
- Create and manage multiple trips
- Automatically converts expenses to home country currency
- Create a travel itinerary and get directions to places when needed

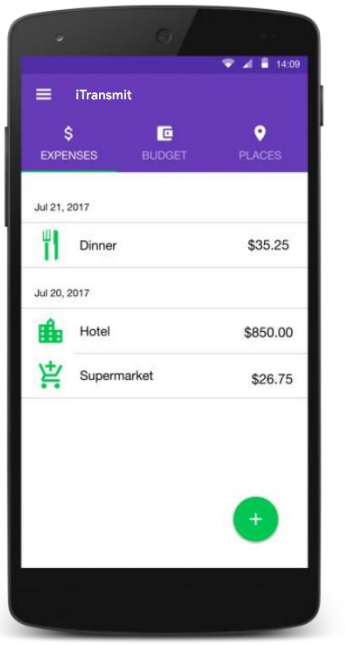
User Interface Mocks

Screen 1 (Sign in screen)

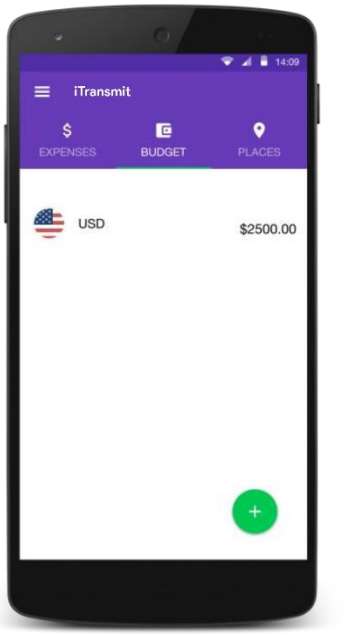


Screen 2 (Trips list)

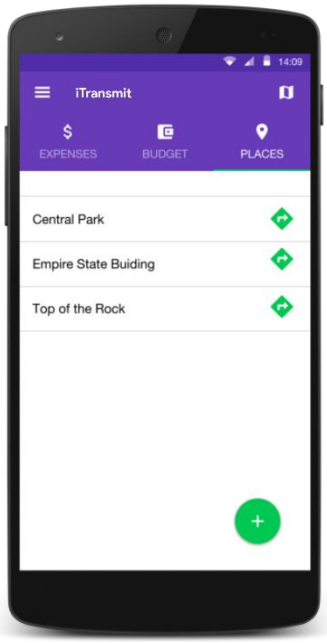
Screen 3 (Expenses list)



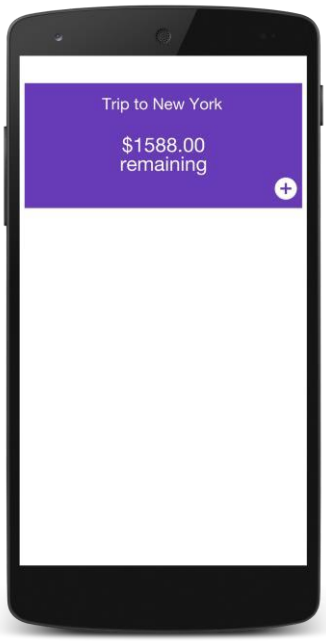
Screen 4 (Budget list)



Screen 5 (Places list)



Screen 6 (Home screen widget)



Key Considerations

- The use only of java for the complete project.
- The use of stable release versions for Android Studio, Gradle and all libraries.
- How will the app support accessibility (content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues).
- Use of `strings.xml` to store all the strings.
- Support for RTL layout switching in all layouts.

Programming Language?

The app will be written solely using the Java language.

Gradle Dependencies?

App utilizes stable release versions of all dependencies, Gradle, and Android Studio

How will your app handle data persistence?

The app will use Firebase Realtime Database to store and retrieve saved data.
This will allow the app to store user data to the cloud, so a simple sign in from a different device would allow it to retrieve all previous saved data

Describe any corner cases in the UX.

The app will have a Navigation Drawer to navigate back to trips list screen and to change application Settings.

The trip detail screen will be divided in 3 equally important parts: Expenses, Budget and Places. Users will be able to quickly navigate between them using a Tab Layout.

In the Settings screen, users will be able to select a default/current trip to avoid the hurdle of selecting it every time the app is opened.

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

- Glide to handle the loading and caching of images presented in the trip list; ButterKnife to better organize layout instantiation; Currencylayer to retrieve real time currencies data.
- Firebase-UI Android to quickly connect common UI elements to Firebase APIs like the Realtime Database and Firebase Authentication.

Describe how you will implement Google Play Services.

This app will implement the following Google Play Services:

- Firebase Realtime Database: Store travel data;
- Firebase Storage: Store expenses receipts or photos;
- Firebase Analytics: Retrieve application usage data;
- Google Places API: Search and store points of interest

Next Steps: Required Tasks

Task 1: Project Setup

- Configure the project by selecting Android target APIs and import libs in the gradle file.
- Generate Firebase and Google Services API keys. Define and set up branch structure on GitHub.

Task 2: Implement UI for Each Activity and Fragment

Implement the UI for each Activity and Fragment of the app:

- Add trips screen and trips list screen
- Add expenses screen and expenses list screen
- Add budget screen and budget list screen
- Add places screen and places list screen
- Settings Screen

Task 3: Implement Material Design patterns

Implement the Navigation Drawer, the Tab Layout and Toolbars. Optimize colors and text spacing.

Task 4: Firebase Realtime Database configuration

Create Firebase Realtime Database scheme and security rules.

Task 5: Usage of Currencylayer API

Use IntentService asynchronously retrieve currencies data to convert expense amount to home country currency.

Task 6: Google Places API configuration

Configure Google Places API to enable points of interests search.

Task 7: Implement home screen widget

Create home screen widget for the app to provide trip's budget information and an easy way to add new expenses

Task 8: Configure installRelease gradle task

Configure installRealease gradle task to make it automatically deploy the app by building and signing it with the provided key.

Task 9: Create usage documentation

Create usage documentation to explain all the necessary information to run and use the application.