GitHub Username: *ashrafsarhan*

Tripto

Description

Finding good trips around the world at a nice price is quite difficult! Tripto makes it easy for you, If you are a couple, travelers, adventures whoever you can use Tripto to find your next destination.

Tripto is a trip deals finder app where you can find, book your trips. Explore among top lists, categories or search your favorites destinations.

Intended User

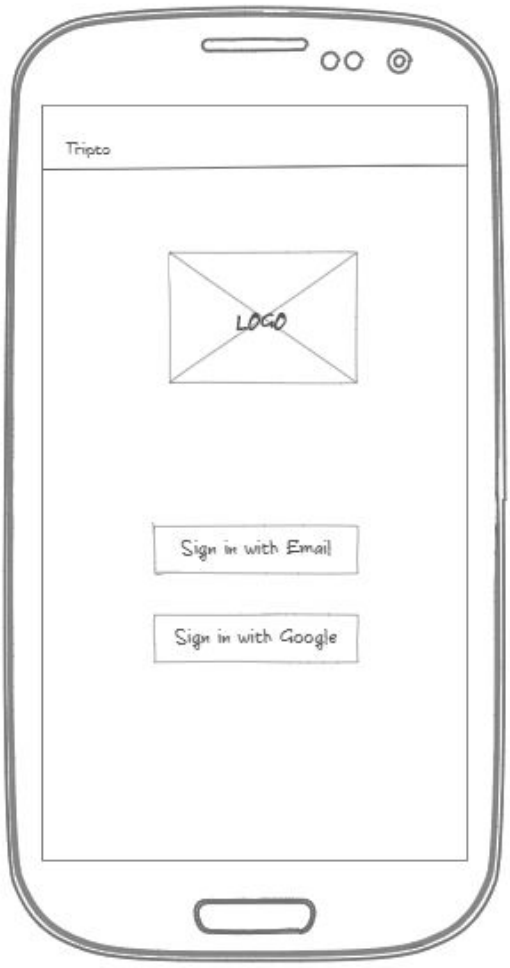
Travelers and everyone who likes to discover the world.

Features

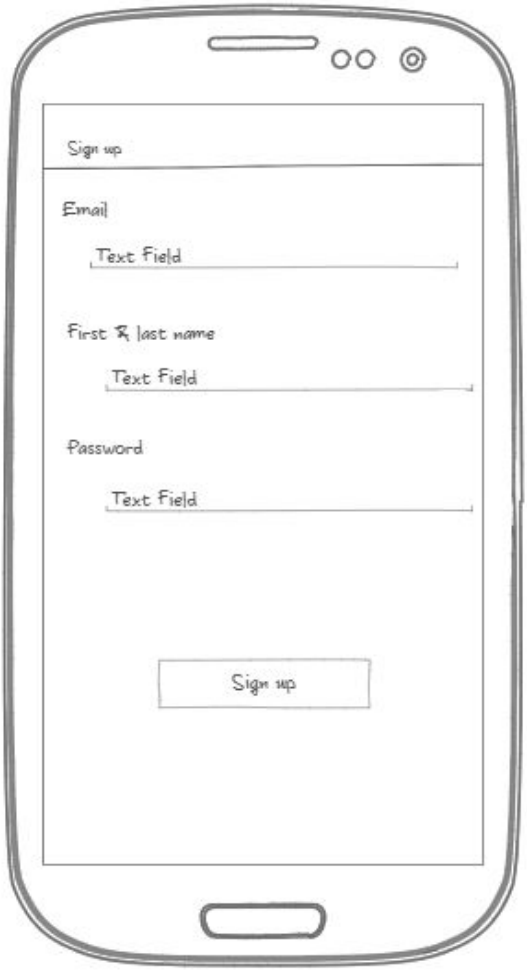
- Sign up
- Sign in
- add/update a new trip (Admin)
- Find and book a trip (User)
- Payment (dummy)

User Interface Mocks

Login

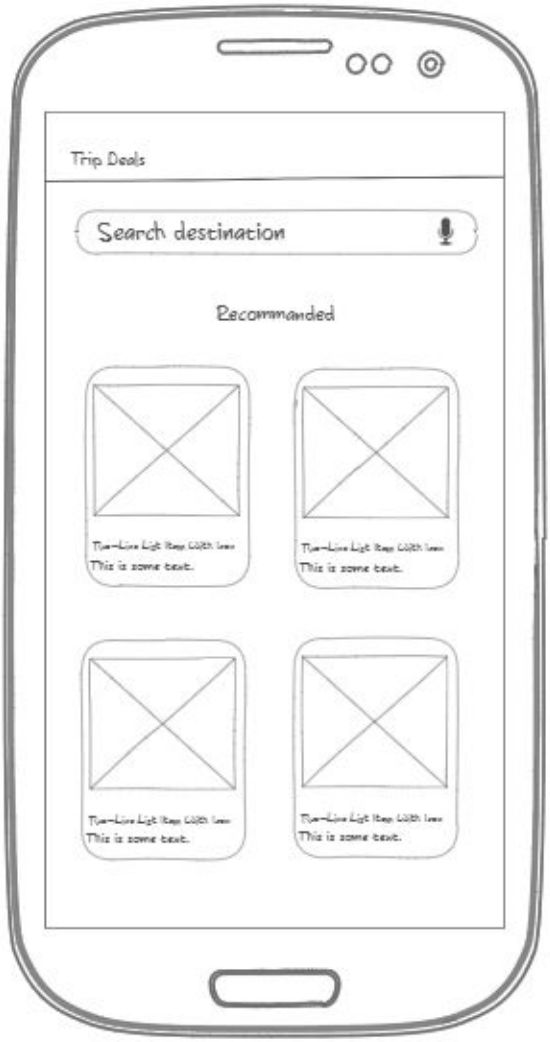


Signup



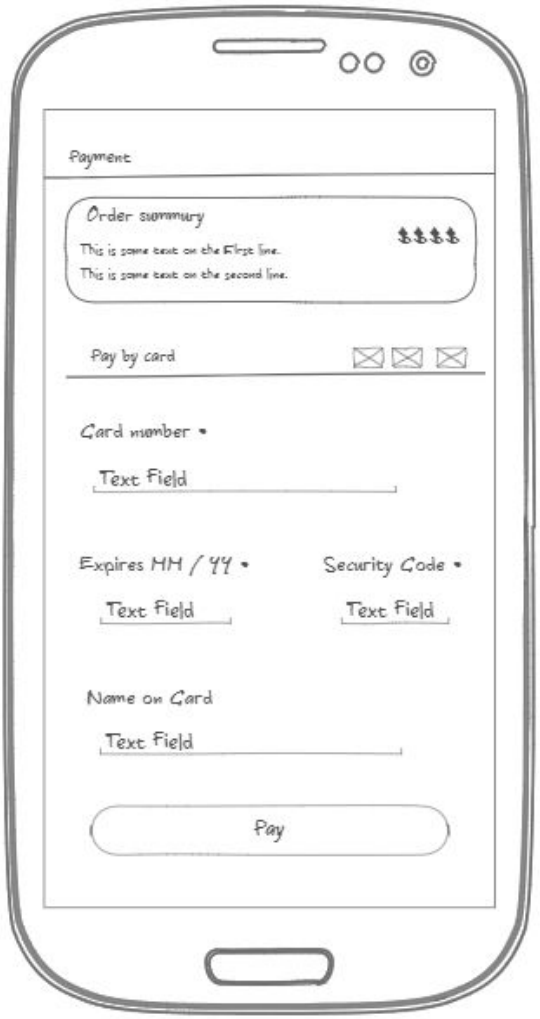
New Deal

Trip Deals



Details

Payment



Key Considerations

How will your app handle data persistence?

Users data including login credentials, their preferences would be handled by Firebase auth API and Firebase Relatime Database API respectively.

Describe any edge or corner cases in the UX.

- The app implementation will try to consider any possible corner case, in order to avoid strange behaviours from the user point of view. For example, when using the app without any network available the local data will still be accessible. And when no data could be retrieved for any reason, placeholder images will be used to point out this situation.
- The possibility of not having any available image for a certain deal will be taken in consideration, in order to avoid strange or broken layouts.
- Handling the authorization for accessing the admin view according to the current logged in user role.

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

- Picasso/UIL - Image loading and caching
- Butterknife - Views Injection
- OkHttp - Handling Networking calls

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

- Firebase Realtime Database - to manage and update real time data across devices.
- Firebase Crash Reports - to observe app behaviour and collect data to prevent crashes.
- Android Design Support Libraries - Material design concepts would be leveraged to provide delightful user experience.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new empty Android Studio project.
- Create a new GitHub repository for the project. Remember including .gitignore and README.md files.
- Configure used libraries (following in each case the pertinent instructions), dependencies and other aspects in the Gradle files.
- Configure the use of Firebase API: get an api key from Google .
- Configure the use of the API key in the project in a way that allows not to include it in the repository.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for main view screen (MainActivity).
- Build UI signin/signup view screens (Firebase UI Integration).
- Build UI for detail page(Deal detail page).
- Build UI for adding new trip deals only for admin users
- All the UI implementation will be made taking into consideration the use of Material Design guidelines, features and components (such as Floating Action Button, Floating Action Menu, Coordinator Layout, Collapsing Toolbar Layout…).

Task 3: Build data persistence support

- Implement all the classes needed to handle data persistence: Content Provider, Cursor Adapters, Database classes, etc

Task 4: Implement Google Play Services/ Firebase integration

- Build Firebase Realtime Database Integration
- Build Firebase Crash Reports Integration

Task 5: Accessibility and localization

- Ensure that the app offers a good enough experience to users with disabilities, through the correct use of content descriptions, consistent and coherent focus navigation, etc.
- Keep all the app strings in XML files, including localized versions for english language (which will be the default language).

Task 6: Configure app building

- Configure app signing, including the keystore and passwords in the repository.
- Ensure that app builds and deploys using the “installRelease” Gradle task.