

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: aldefy

NextStop

Description

Discover and plan your next trip to the top destinations. Explore the world and get inspired to travel more with this travel app. Get personalised recommendations of things to do, places to see.

Intended User

NextStop is intended for users who are looking to explore new places to visit , plan a holiday or share travel plans with friends .

Features

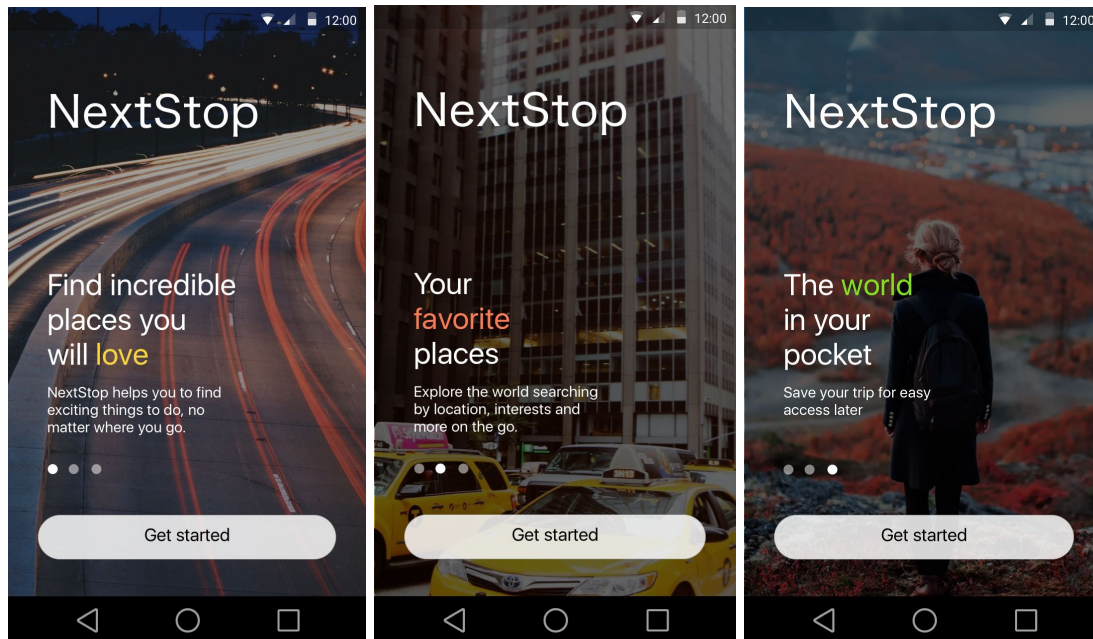
- Recommends top places to visit - uses Google Places API
- Personalized user experience based on user's profile , the app lists the top places a user would be interested to visit next.
- Search for places by keywords , natural language.
- See the details of a certain place.
- Mark and plan a trip with friends
- See Flight info , accommodations , custom tour package information
- Receive recommendations from friends.
- Personally configure some application settings: default order used in the results screen, mark trip dates , organize a chat with travel buddies.

Project Stages

- Stage #1 - Build the core / skeleton app with the list of places and recommendations and their detail screens
- Stage #2 - Build a improved search functionality and load data dynamically based on selections and user's data usage history.
- Stage #3 - Add functions to allow user to invite his friends / family , share some data on Social networks with peers.
- Stage #4 - Add feature to plan a trip and book - fetch trip cost , accommodations and build a custom trip for the user to share with friends, this includes sharing data to calendar api and enabling user to book in-app (using Skyscanner apis)

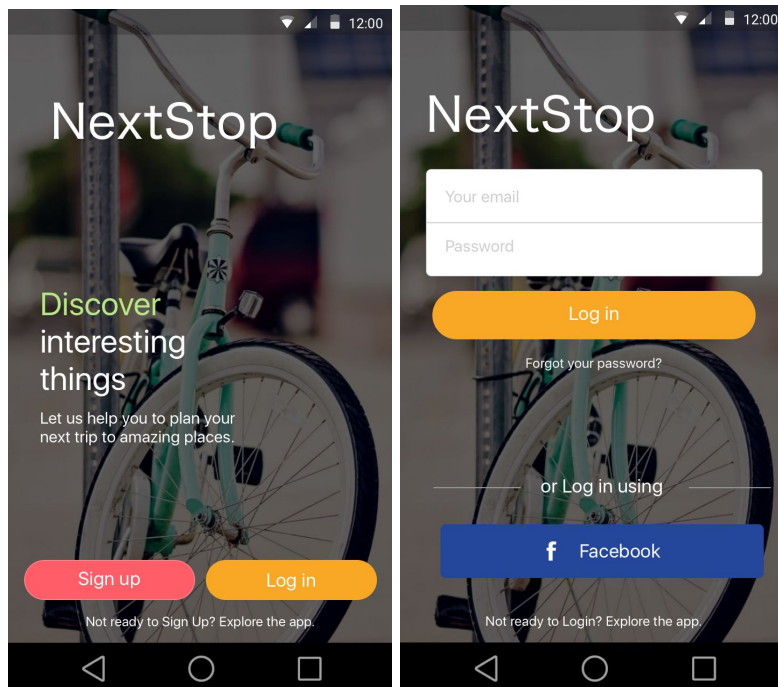
User Interface Mockups - Sketches

Splash walk through screens



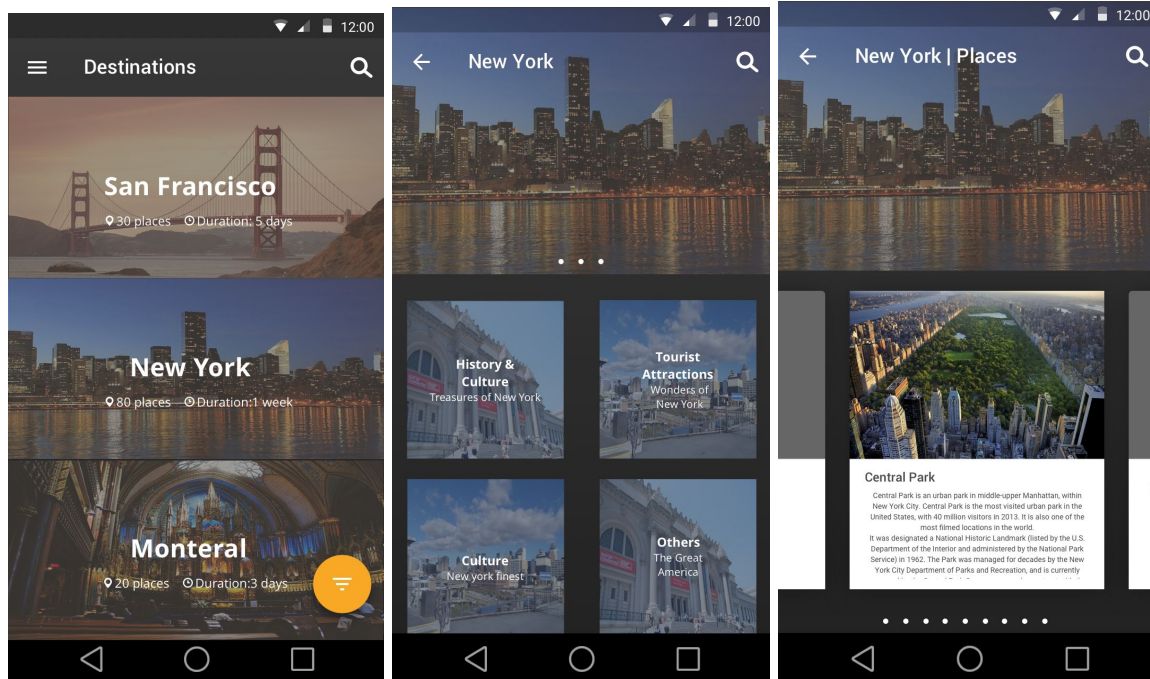
Splash screens - Viewpager slideouts , same android activity but 3 different auto scroll images and app descriptions. This screen gives a gist of the app primary features.

Onboarding user



Onboarding screens - the screens will also build a user profile based on few selections.

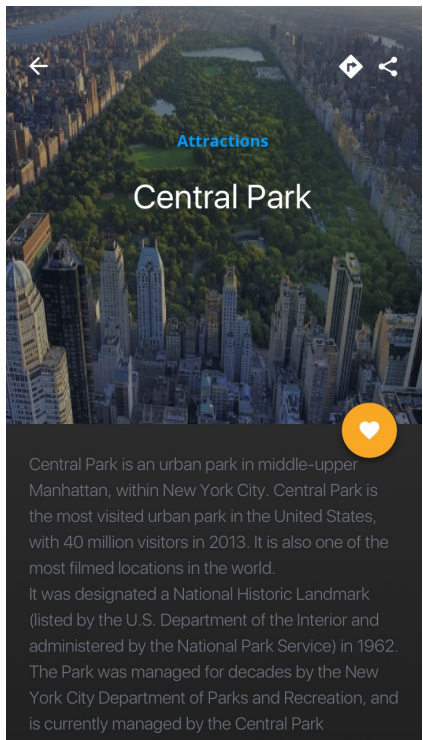
Main app



Left to Right

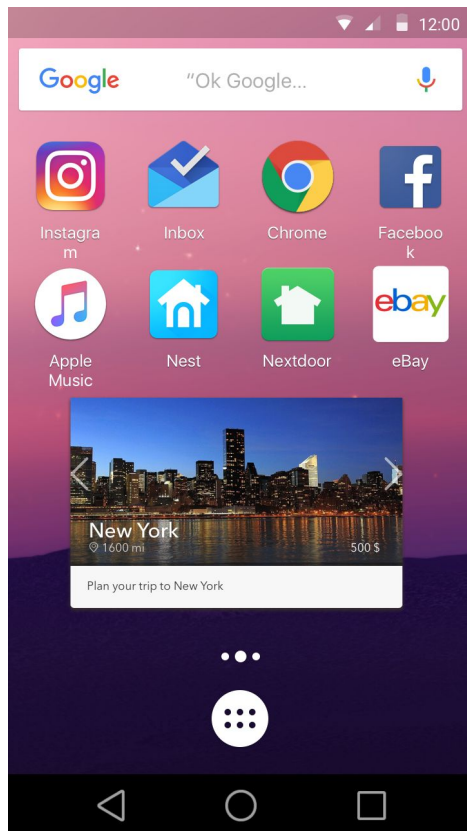
- HomePage : List of places recommend places / cities - this is populated based on user profile and user details during signup , ex - beaches , adventure , solo , romantic etc , list is auto populated to best recommend the top destinations user would likely be interested to attend.
- City detail page - A curated list of top attractions / history and info about the place split into multiple categories
- Individual category - Ex : Tourist Attractions of New York

Details Screen



Detail screen for a particular place , user can read advance in depth details of a particular attraction , mark as fav , plan a trip , recommendations and similar attractions are suggested.

App Widget



Widget features :

- Auto generates a new place everyday and lists some details(phase 1) , user can then tap on it to be directly taken to the city details page with relevant info.
- Skip a city by navigating to prev or next city in a list.
- Resizable

Key Considerations

How will your app handle data persistence?

App fetches the initial data of recommended cities / things to do (Generic irrespective of a particular city / state / country / place) from the Google Places api and caches the data locally into SQLite using Content providers and loaders

- List of data is fetched based on different parameters - such as user's location,etc
- Caches the top 10 cities and few sub level details into a DB structure
- Save user information and fetch for later use [User favourite his / her places etc]
- Provide search based on pre cached data (cities and their data)

Describe any corner cases in the UX.

Stage # 2-4 in the app use - will provide user with a UX flow that enables him / her to organize and plan a trip , the user can loose track of information they need to plan end to end trip .

This particular case would enable me to deliver targeted push notifications to bring back the user to the app , enable the user to generate local data as per his selection and build a better UX experience based on the interactions user generates on each feature.

App would then pop reminders and relevant booking information required for the users to complete the trip planning and recommend places based on their interest.

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

- Glide - Image caching and building a smooth UX experience throughout app with low memory consumption
- Google Play services - FCM , Places api , maps , location api - to deliver rich content to user at different points of interest in the app.
- Dagger - Dependency injection to generate and provide runtime and singleton dependencies
- Butterknife - Resource dependencies injection
- RxJava 2 - Improve on background processing of data and handle threading
- Retrolambda - Using lambda functions of Java 8 to reduce boilerplate code
- Logan Square - JSON parsing , faster than gson
- Retrofit 2 and OkHttp - Network layer and communicating with different api services.
- Firebase Database - User profiles and data
- Support libs - Material design and annotations
- Social integration such as Facebook sdk , fabric for crashlytics

Describe how you will implement Google Play Services.

Google Play services lib - Places , location , maps

Firebase push notifications and database and authentication libraries

Play services gradle plugin - 9.4.0

Next Steps: Required Tasks

Task 1: Project Setup

- Build a Skeleton project structure with required libraries
- Setup base package to implement MVP practices
- Configure the dependencies
- Declare any static constants
- Call out the base package structure and feature sets

Task 2: Onboarding flow

- Integrate Firebase and Facebook login flow with onboarding process for user to build a profile or login
- Build a UI flow which builds a user profile and generate a recommendation data set
- Create UI layout files with corresponding activities / fragments for the onboarding flow.

Task 3: Fetch and cache data

- Call relevant api to fetch data - Locations and Places api
- Generate POJO / models for the database structure.
- Build a image cache dir and initialize the network layer

Task 4: Architect the backend on Firebase

- List out the data that firebase database will store and deliver to the app for personalized experience of individual user.
- Build mechanism to recommend places based on user data
- Build a contract between Firebase and app for FCM / database.

Task 5: Phase #1

In phase 1 of the app , the app will enlist the top recommendations for user

- Setup homepage lists and sorting functionality
- Add a Search
- Handle the app overall navigation flow - Navigation drawer

- Generate relevant adapter and placeholders fields.
- Build and connect the network layer with the data models and display the data on recyclerview.
- Add a sorting functionality
- Design and build a functionality to enable user to favourite places and browse different categories
 - Connect the list with its details page - City level 1
 - Connect the individual categories / data points to the Details page - City level 2
- Build a user profile screen for user to see his user details and access his favourite list.

Task 5: Phase #2 ,#3 and #4

- Build a social platform for users to share their content to friends.
- Plan a trip
- Book flights / accommodations based on the user selection.
- Build a backend service to trigger push notifications

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"