GitHub Username: nastjashevchenko

Discovery Time

Description

Discovery Time is application to make your travel planning clear and painless. We all love travel, OK, not all, but most of us. Imagine you are going to visit the country you dreamed about for your whole life. You want to see a lot of attractions and don't miss anything, just put all sights to your trip and organize it day by day, spending time wisely, deciding where to stay en route and keeping useful notes. Not ready to travel right now, but heard about great place you must see before you die? Just put it on a wishlist and do it later!

Intended User

This application is great for (of course) for world discoverers: everybody who loves discover new places in distant lands, can't live without adventures or just wants to have a rest with family on weekend not so far from home.

Features

- Save your travel ideas in a wishlist. You won't forget about recommendation friend gave you or some great place you have read about in a magazine. Anytime here are holidays, vacation or just weekend you don't need to think what to do, just check your list!
- Add ideas of must-see and nice-to-see places for one trip, check how far they are from each other to plan itinerary and places to stay wisely, don't waste your time driving in circles!
- Collect useful notes and tips, keep additional information like plane schedule, booking numbers or host contacts.

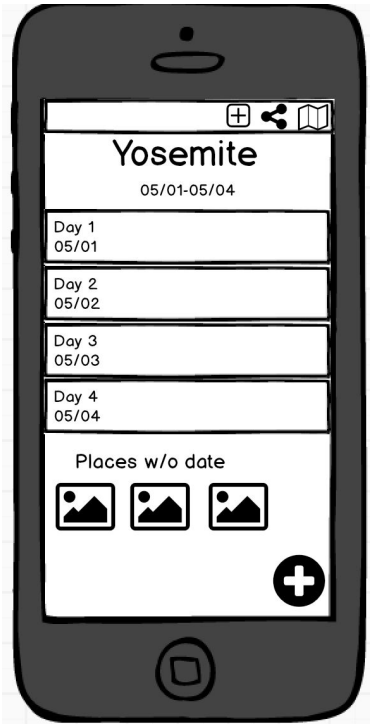
User Interface Mocks

I didn't include any add/edit dialogs app would include, but generally this functionality would be clear from screens descriptions.

Screen 1 - List of trips

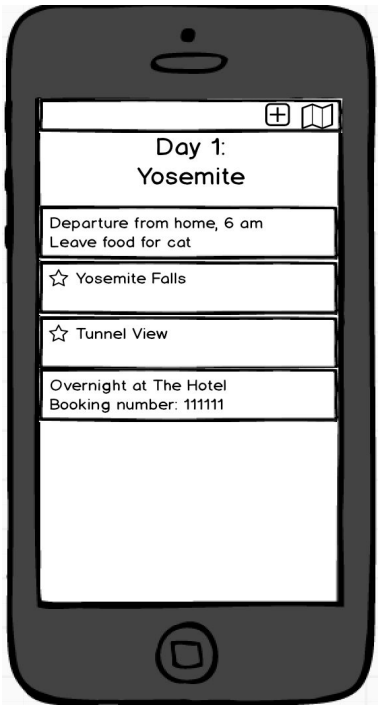
The main screen is a list of all trips, it can contain upcoming trips or just ideas for trips. The logic is that if dates are added then a trip considered as planned, otherwise it goes to wishlist. There should be one more sublist for past trips. The alternative to what is shown on screen - tabs for each list at the top.

Screen 2 - Trip details



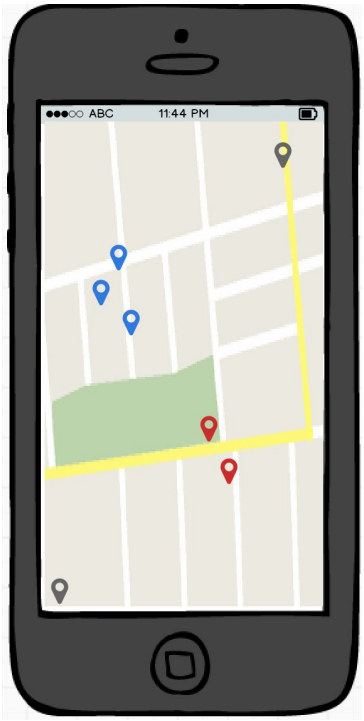
User can click on trip and open trip activity. It has as many days as set by start and end date for day-to-day planning and notes. Also trip has places to visit, which can be added to particular date or without date. If it is added w/o date it shown in separate list under all days. All places for the trip can be shown on a map by clicking map icon in toolbar.
Use case: I want to visit bunch of particular places, but want to put ones that close to each other to one day. I add everything I want w/o date, look for it on map and then assign each to day when I know which “order” is better and have approximate itinerary. Also this process can help to understand in what cities/districts I want to look for accommodation.

Screen 3 - Day details



You can click on item in days list and open day details. Each day can contain 2 types of elements: place to visit (which is shown on map) or text note with any tips or details you want to keep for this day. I marked places with stars just to make it more clear on mock that here are two types of items.

Screen 4 - Map example



From trip or day screen map with all places can be opened. I am going to group markers by days using color. E.g. on the screen I plan to visit blue ones during the first day, red ones - the second day. I am not sure with greys, because it is pretty far and I need to plan additional time to commute.

Key Considerations

How will your app handle data persistence?

I am going to store trips and trip data in SQLite DB. I would prefer communicating thru some ORM, but as far as Content Provider is requirement for this project I would create one (although I think it is redundant for this application).

Describe any corner cases in the UX.

- Empty screens: when we have no trips added or we haven't added anything to day plans
- If our trip is one day (I just want to visit some place relatively close to my home) and dates cover only one day I don't need list of days and can merge trip and day activities into one
- When dates are changed so that days number is changed. Especially when something is already added to deleted day.

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

Not sure for now what else can be used, but I am pretty sure I will use some library to load and cache images like Picasso or similar, Butterknife, because it makes life easier when activities/fragments contain many views. I also will use Google Places/Maps APIs and widgets to search for places and show on map.

Next Steps: Required Tasks

This is the section where I take the main features of the app and decompose them into top level tasks to work on.

Task 1: Project Setup

Create new project, I will add libraries when I need to use them in code.

Task 2: Implement UI for Each Activity and Fragment

Implement UI layouts and classes with mock hardcoded data for:

- MainActivity for trips lists (wishlist, upcoming, past);
- Build Activities and Fragments for trip details and day details screens.

Task 3: Create model and ContentProvider

- Create classes for each object: trip, day, place. Likely, objects should be Parcelable;
- Create Content Provider and underlying DB to query, update, add and delete entries.

Task 4: Create add/edit dialog and other actions like:

- For trip: add/edit title, dates, places;
- For day: add note, add place;
- For place: find new place, add to trip (w/o date), assign to day.

Task 5: Implement Google Play Services

- Google Maps to show markers on map;
- Google Places to add place picker or autocomplete finder to add places;
- Google Analytics to track users activity on different screens.

Task 6: Add widgets

- Simple widget with "X days to your next trip";
- List widget with upcoming trips

Task 7: Add LTR and accessibility

- Enable RTL and check all screens are OK for layout mirroring;
- Add content descriptions;
- Check all strings are in strings.xml.

Task 8: Design

- Add application there with material palette;
- Add nice toolbars, e.g. collapsing toolbar for trip screen with picture at the top;
- Add animated transitions between screens.

Task 9: Add signing configuration

- Add configuration and keystore.