GitHub Username: narko

TourO

Description

TourO is a mobile app that helps discovering UNESCO world heritage tourist attractions and must go places in an easy way. Using the same swiping mechanism implemented by Tinder, the app allows users to navigate a list of points of interests (POIs) and get relevant information about them.

Just like in Tinder, swiping right or left means the user likes or dislikes the POI depicted on the screen. When a user likes a POI, this is automatically added to a list of favorites.

Intended User

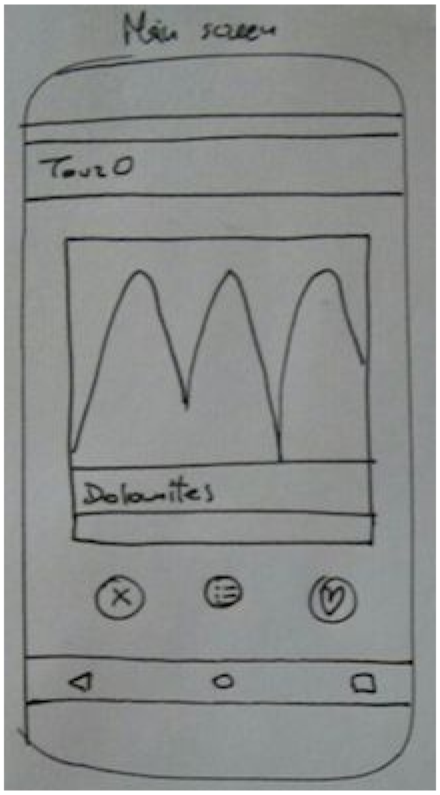
This app is for travelers willing to see and discover well-known places on their own.

Features

- Navigate list of POIs in a Tinder-like style.
- Save POIs to favorites.
- Show information about each POI with available videos, photos and description. Note that the multimedia content depends on its availability for each POI.
- Allow the navigation of existing POIs in the favorite list.
- Remove POIs from favorite list.
- Display list of POIs on a Map.

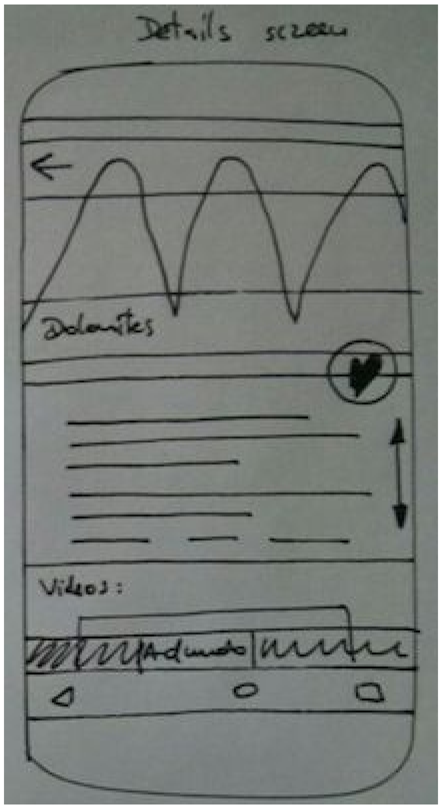
User Interface Mocks

Screen 1



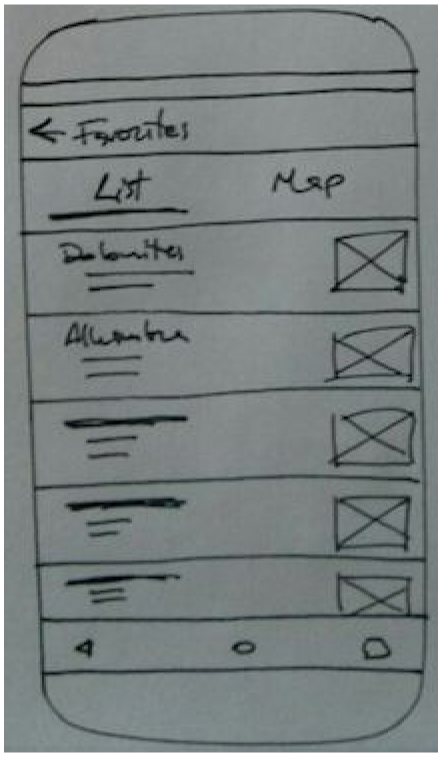
Main screen showing Tinder-like card navigation. Users can swipe left (dislike) or right (like). The same functionality is also achieved by using the fabs from the bottom. The fab in the middle opens the list of favorite POIs. Tapping on a card will display the details screen for that specific POI.

Screen 2



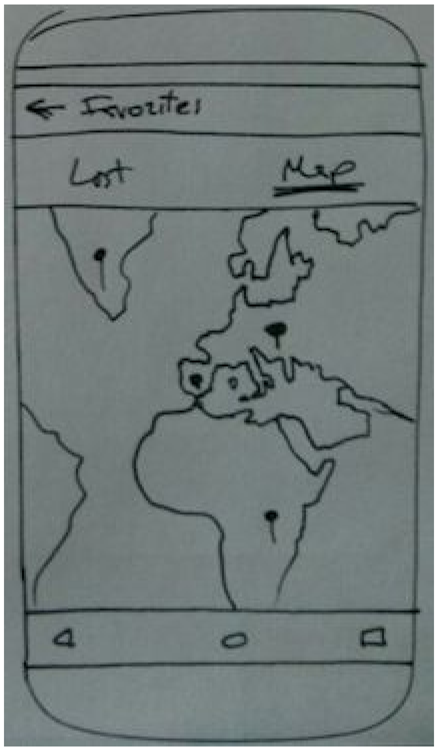
Details screen showing specific information about a POI: feature photo, text description and videos when available. This screen allows users to add / remove the POI to the favorite list by pressing the FAB.

Screen 3



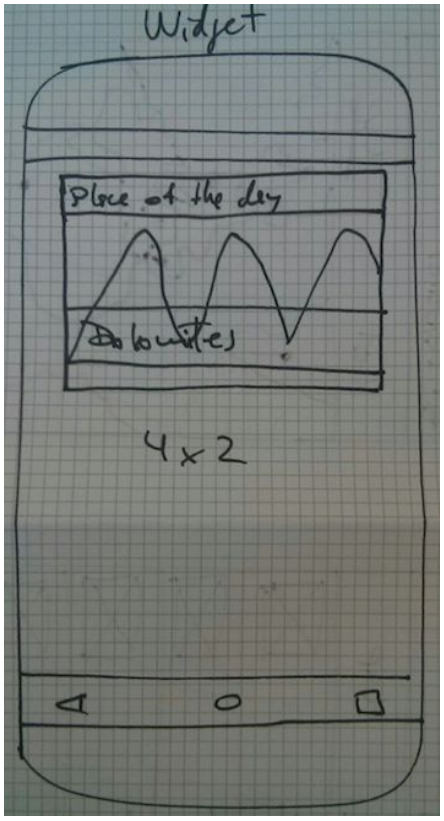
Favorite screen (list tab): this screen shows the list of saved POIs. A tap on an item will display the details screen. It is possible to delete an item on the list by just swiping.

Screen 4



Favorite screen (map tab): this screen displays the list of saved POI on a Google Map.

Widget



The widget will show a random place for the given date. The layout will contain a featured photo together with the name of the place. By tapping on the photo the details screen will be shown to the user.

Key Considerations

How will your app handle data persistence?

The data will be stored locally in SQLite by using a Content Provider. The Content Provider will be implemented manually from scratch.

Describe any edge or corner cases in the UX.

One important issue is how to avoid showing the same POI in the available card deck once the user navigates back from a detail screen. The app should control this situation by implementing some kind of randomization for presenting the list of available POIs to the user.
In addition, we should take care of not showing and storing duplicated POIs in the favorite list.

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

- Picasso will be used for loading images.
- Constraint layout will be used for creating the UI when possible.
- Butterknife will be used to avoid boilerplate code when inflating views.
- OkHttp will be used for handling the HTTP communication.

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

The app will integrate the following services:

- Google Maps: this will be necessary to show the list of favorite POIs.
- Analytics: this will be needed in order to get more insights about the behavior of users of this app.
- Admob: a little add will be shown in the detail screen.
- Data service: this is the service providing the JSON data. The consumption of this data will be done by using OkHttp. The data will be synchronized with the local database by using an IntentService.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create a new Android project.
- Configure build.gradle with the required dependencies.
- Setup git repo and make an initial commit.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity (Tinder-like navigation)
- Build UI for POI details.
- Build UI for POI favorite list.
- Build UI for POI favorite list depicted on a map.

- All screens showing data retrieved from the Content Provider will rely on a CursorLoader for doing so.

Task 3: Data preparation

This task consists on scraping Wikipedia and/or other data sources for retrieving the list of all UNESCO World Heritage places. This data will be modeled in JSON and stored on Firebase.

Task 4: Remote data request

- Implement HTTP request to the online data store.
- Parse JSON data.
- Populate UI with the retrieved data.

Task 5: Local data store

- Build Content Provider.
- Add tests to check that the SQLite DB works as expected.

Task 6: Google Play Services integration

- Implement Google Maps functionality.
- Implement Analytics.
- Add admob functionality to POI details screen.

Task 7: Control error-prone situations

- Control the case when Internet is not available and adapt UI.
- Control the case when the list of favorite POIs is empty.

Task 8: Widget

- Widget layout.
- Widget intent and navigation implementation.