GitHub Username: Barrick89

PerfectWeekend

Description

We all know this situation on weekends: The sun is shining and the birds are singing so it's the perfect time to do a trip with friends or family. But where should you go? How about a picnic at this beautiful lake in this nice little town you were a few years ago? Or how about this cute art café with the delicious chocolate cappuccino you were last year? But how was the cafè called? You can't remember the name of the little town either not to mention finding this perfect picnic spot at the lake you were last time.

“PerfectWeekend” can help! Everytime you find a perfect spot that you want to visit again in the future, just add it to “PerfectWeekend” by taking a photo and adding a short title. You can even add longer notes if you want to! The app will save the coordinates of the location for you so that you always know how to get back to your favorite spot and you can also browse through an album of all the places you added to pick a couple of spots for your next perfect weekend. Where other apps set the focus on presenting new places to the user or sharing certain locations with others, “PerfectWeekend” specializes in preserving your favorite spots.

Intended User

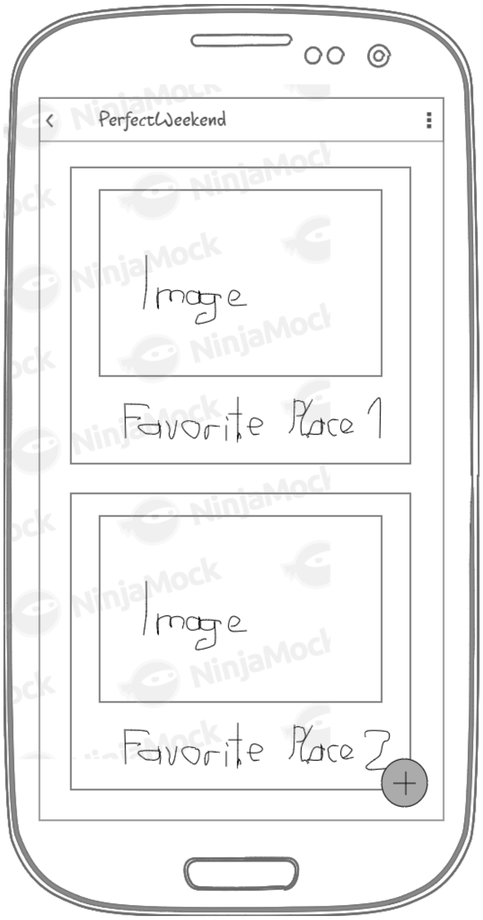
Everyone who likes to spend their free time outside and not at home. It might be especially suited for families to keep track of places the kids enjoyed most. It might also encourage couch potatoes, who don’t like to go to new places, to go out to their favorite places more often.

Features

- saves information (image, title, description, coordinates) for a chosen location
- lists the saved information
- shows a location on a map
- provides a widget which contains a minimalized version of the favorite location list

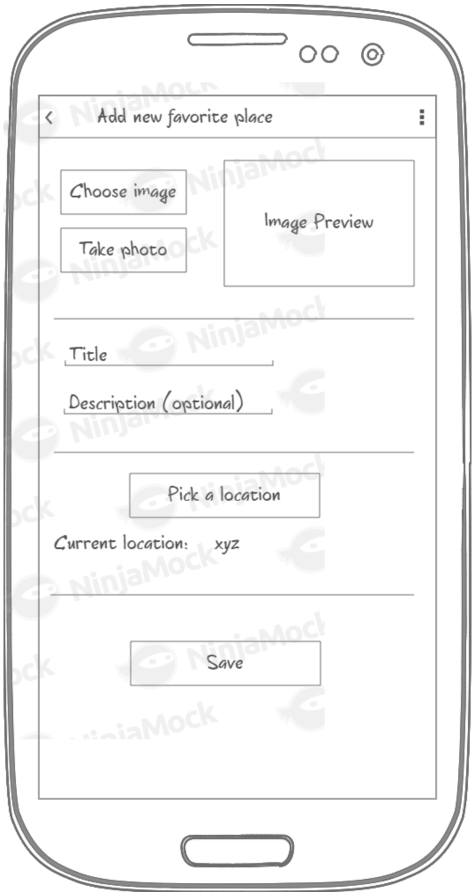
User Interface Mocks

Screen 1



MainActivity contains a Recyclerview with Cardviews. Each card shows the image and the name of a favorite place that the user added. When the user taps on a card the DetailActivity opens. When the user taps on the FAB the EditorActivity opens.

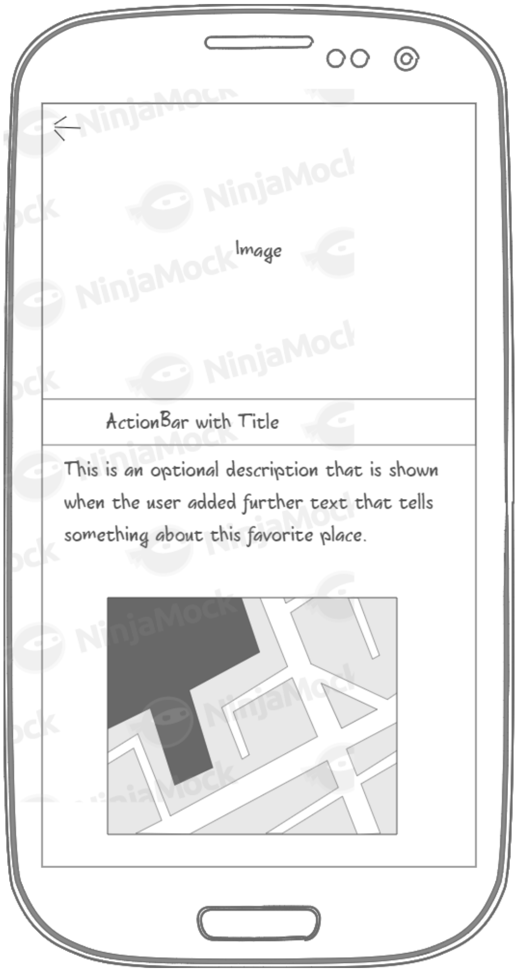
Screen 2



In the EditorActivity the user can add a new location. First he can decide whether to use the button “Choose image” to use a picture from his phones data storage or to use the “Take photo” button to take a new picture. A preview of the image is then shown.

Then the user has to add a title and optionally can write an additional description. The last step before hitting the “Save” button is to pick a location by hitting the “Pick a location” button. The chosen location is then shown. If the user tries to save and necessary informations are missing, a toast will appear and tell the user to complete the informations.

Screen 3



This is an optional description that is shown when the user added further text that tells something about this favorite place.

The DetailActivity shows all the informations that were added by the user in the EditorActivity including a map where the favorite location is marked. The user can access DetailActivity by tapping on a list item in MainActivity.

Key Considerations

How will your app handle data persistence?

The app uses a content provider and a SQLite database to store the information that the user adds.

Describe any edge or corner cases in the UX.

A edge case in the UX could occur when the user forgets to save the location to his favorites while he is there. The solution would be to give the user the possibility to choose a location on the map to add it to the favorites list.

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

- "Picasso" to handle the loading and caching of images because the app uses photos that the user took
- "Butterknife" to simplify referencing to UI objects

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

The app uses the location and maps services. Location is used to save the current location to the favorite list. Maps is used to view the location in the detail screen.

Next Steps: Required Tasks

Task 1: Project Setup

- Add Picasso and Butterknife dependencies to the Gradle files
- Add the Google Maps and Location service dependencies to the Gradle files

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for EditorActivity

Task 3: Implement data storage

- Implement contract class
- Implement SQLiteOpenHelper class
- Implement content provider

Task 4: Implement MainActivity

- Implement cursor loader
- Add an adapter class to provide the recyclerview with data

Task 5: Implement EditorActivity

- Implement the option to pick and save a location with Google location services
- Implement the option to take/pick a photo
- Implement the option to add a title/description

Task 6: Implement DetailActivity

- Implement the code for showing the image and the text
- Implement a map with Google Maps Services

Task 7: Implement widget

- Implement an AppWidgetProvider class
- Implement a RemoteWidgetService class