Proposal Contents

GitHub Username: Draturan

MyLittleJourney

Description

MyLittleJourney is an App for travelers, thought for people that want to write down every little things happened during the trip, designed to be simple and easy to use, like a diary!
Have you ever read about something you wrote in a quite far past, and then you realize that you forgot a lot of little particulars of that experience?
MyLittleJourney wants to help you save them!

Intended User

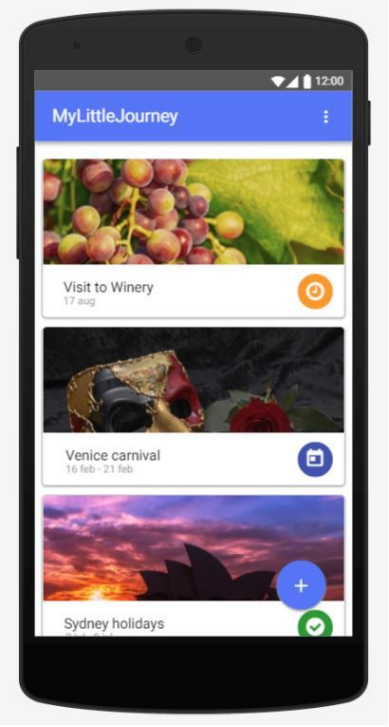
The intended user is a traveler who likes to write down a travel diary, to remember, maintain and tell to family members or friends all the adventures encounter in his/her journey… or maybe just to don't forget them!

Features

- Add or remove Journey
- Write general information of the Journey
- Write a diary
- Reminder if was forgotten to write during the day

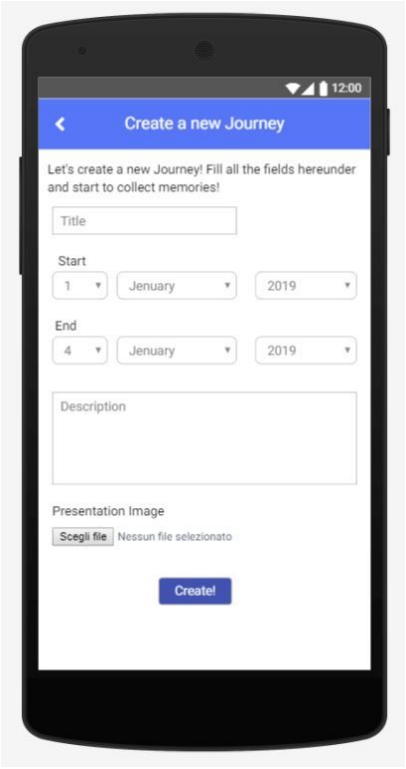
User Interface Mocks

Main Activity



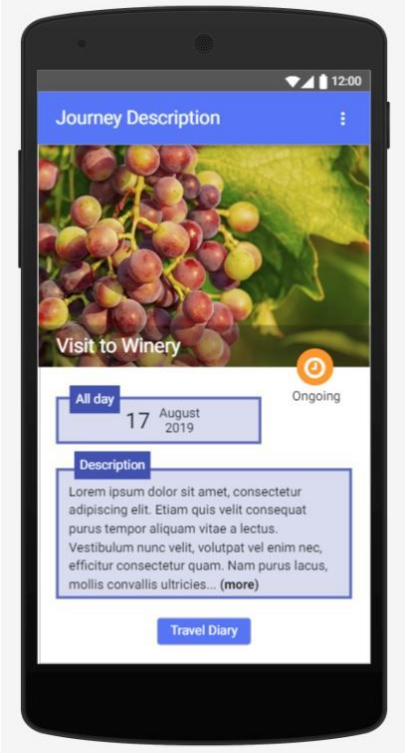
The user will find a little suggestion to start with the first journey if there are no Journeys present in the database. Otherwise the journey will be divided into "ongoing", "programmed", "completed" status.

Add Journey Activity



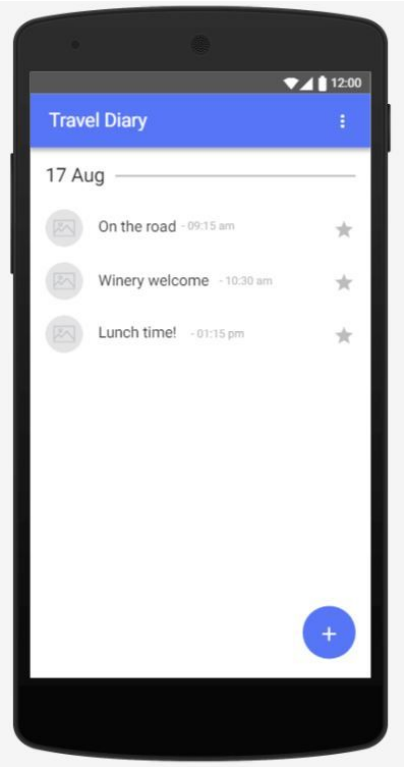
By clicking on the Add FAB the user will be bring to **Create a new Journey** Activity, where he can fill all the field and finally create the item.

Journey Description



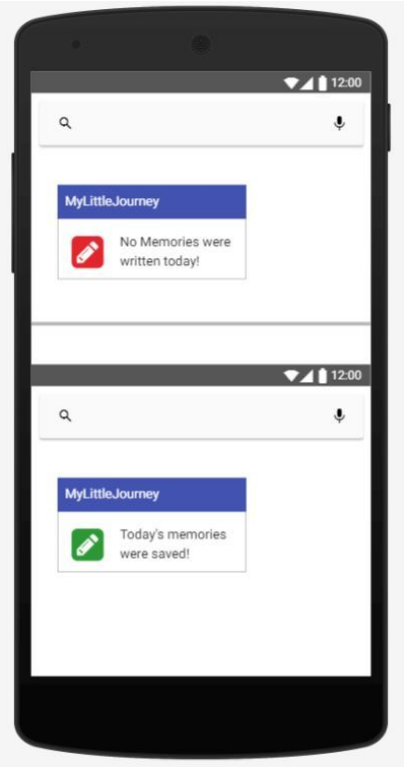
When a user click on a Journey item in Main Activity will see all the other information of the Journey like Description that was hidden on main view and the button to go to **Travel Diary Activity** to start adding new trip memories.

Travel Diary Activity



Finally clicking on “Travel Diary” button the user go to the activity where will be able to write everything is happened during the current date.

Widget



The Widget suggest or remember to User if the memory of the day was written, but only if there is an ongoing Journey.

Key Considerations

How will your app handle data persistence?

The usage of Firebase Realtime Database will provide to App needed feature of data persistence, in both Local and online situation.
To avoid data losses on device rotation or destroying of an Activity all needed data will be saved with onSaveInstanceState method.

Describe any edge or corner cases in the UX.

If the User fill all the fields of an adding Journey or an Add Memory the records will be not saved until “Create!” or “Save” button was pressed, so in case of returning back the previous Activity data should be written again. A message of data losses in case of returning back could appear to ask the User if is really sure. In case of a connection absence (that may happen when we are in a LittleJourney) the data is recorded in the offline database and synchronized when a connection is retrieved.

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

- **Glide:4.7.1** to handle the loading and caching of images
- **Butterknife:8.8.1** to avoid, where possible, usage of findViewById() method
- **Firebase:** to handle database and data persistence, authentication and analytics

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

- **Analytics:16.0.1** in a single instance, will help to better understand the usage of the app to improve further updates
- **Authentication:16.0.2** to provide the user with a *uid* that is related to a single Database node, and so a personal list of journeys
- **Database:16.0.1** to handle data persistence and information
- **Storage:16.0.1** to handle pictures

Next Steps: Required Tasks

Task 1: Project Setup

The project will be written in solely Java Programming Language. The development will be made on **Android Studio 3.1.3**, compiling work will be entrusted to **Gradle 4.4**.
The **minSdkVersion** will be **16** and the **targetSdkVersion** will be **27**. App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts. All images will have a contentDescription to enhance accessibility and support for D-pad navigation.

List of subtasks:

- Configure libraries
- Build App Activity skeleton
- Implement Anonymously Authentication
- Implement logic for database
- Implement Authentication and Database controls
- Implement Storage and his logic
- Implement Analytics and first data check
- Implement widget with an IntentService
- Improve graphic interface for a better UI experience

Task 2: Implement UI for Each Activity and Fragment

List of subtasks:

- Build UI for MainActivity
- Build UI for JourneyDescription
- Build UI for AddJourneyActivity
- Build UI for TravelDiaryActivity
- Build UI for AddMemoryActivity
- Build UI for Widget

Task 3: Implement Services

List of subtasks:

- Implementation of Firebase Authentication
- Implementation of Database and Storage
- Implementation of Analytics
- Implementation of Glide and handling of present images

Task 4: Handling Errors

List of subtasks:

- Handle errors caused from retrieving data
- Handle errors caused by images unretrieved data
- Handle errors caused by low or null internet connection

Task 5: Widget

List of subtasks:

- Implementation of Widget UI
- Implement communication through an IntentService
- Build and check the functionality

Task 6: Completion

List of subtasks:

- Check UI usability
- Lasts check on UX border and corner cases