GitHub Username: vjauckus

Personal Diary

Description

Save secured personal daily data in diary / journal using firebase auth, realtime database and storage.

Intended User

Intended user are people all ages, who like memorize own experience and mood.

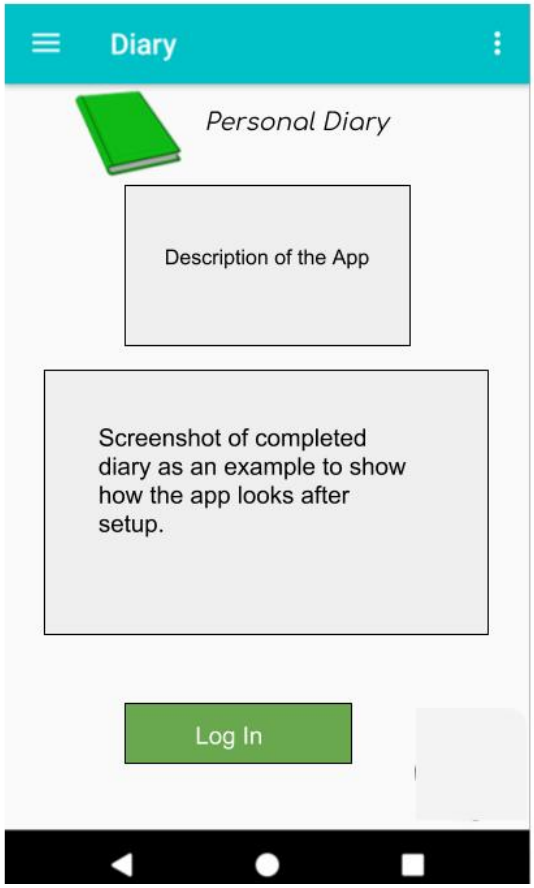
Features

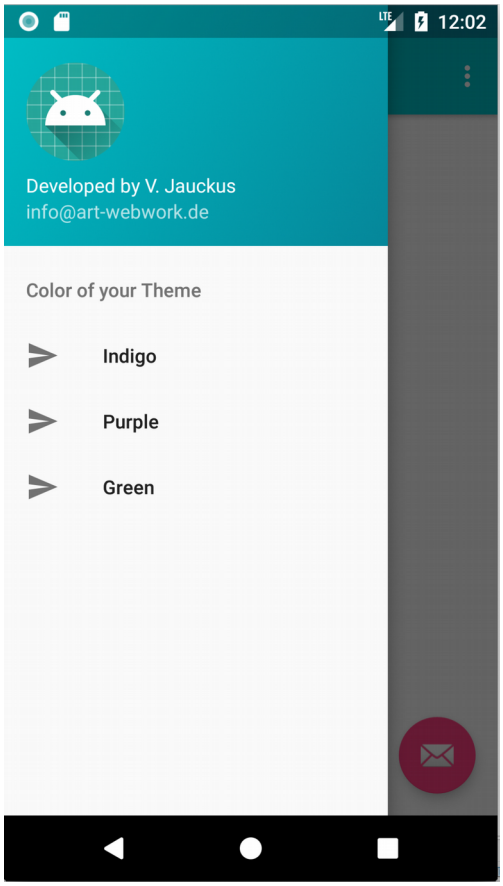
List the main features of the “Diary”app are:

- Saves information secured
- Memorize pictures
- Sorting the entries
- Have a fun

User Interface Mocks

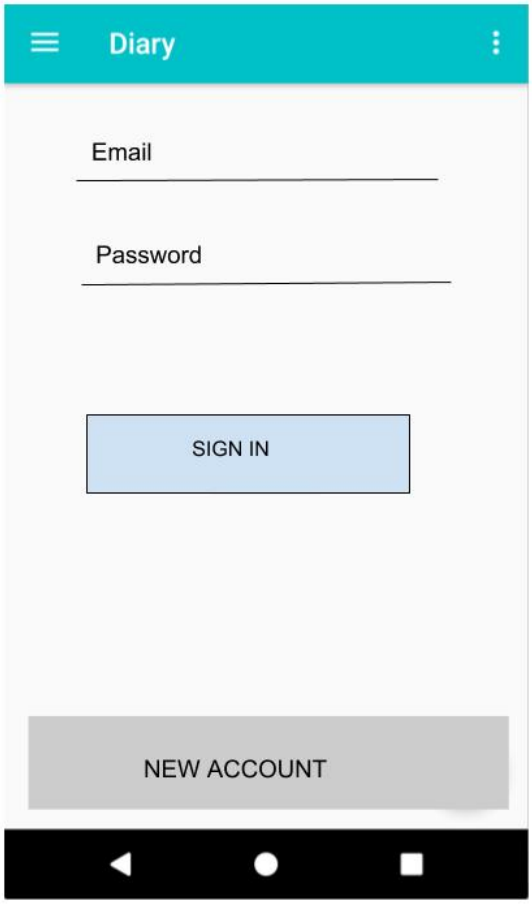
Screen 1





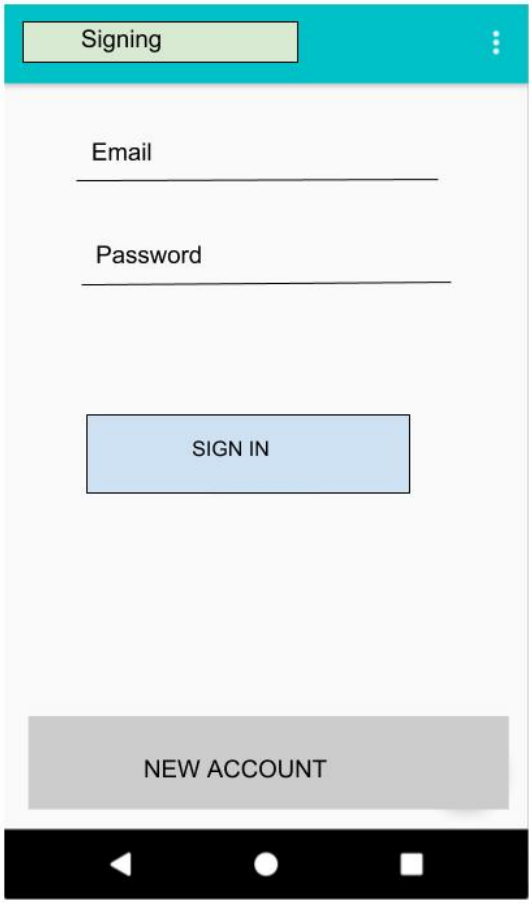
The first screen helps the user to understand what the App is about. The screen shot shows a completed diary as inspiration. Use sandwich menu on the left to change the colors of the theme to make the journal more individual.

Screen 2



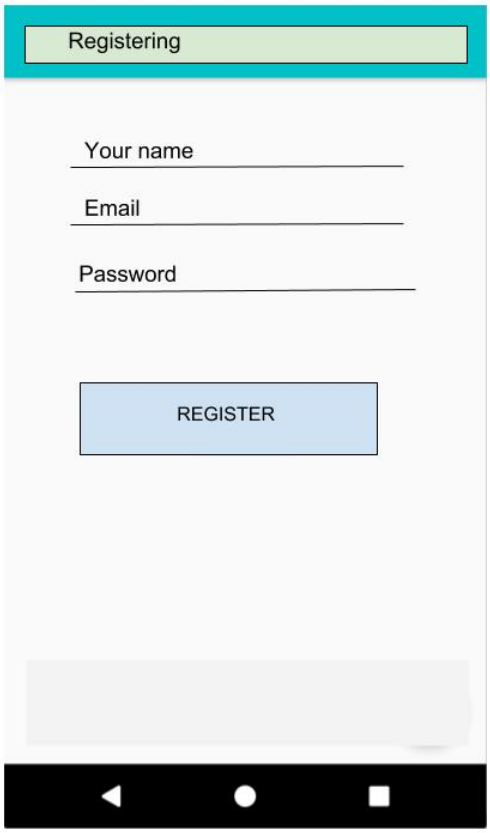
Sign in, if you already have an account or register. Use the menu on the right to sign out.

Screen 3



Sign in, if you already have an account or register. Use the menu on the right to sign out.

Screen 4



This is the register screen for first usage of journal.

Screen 5

This screen is for input daily impressions: positive and less positive in text form.

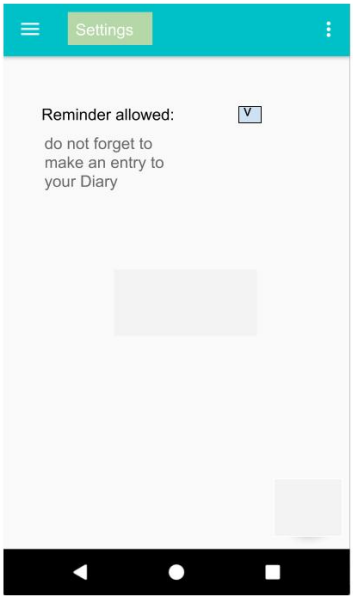
Additional the user can put an image and give a vote for whole day's mood.

Screen 6

This screen provides the list of all saved entries. The menu allows to sort data by best estimation and show favorite days.

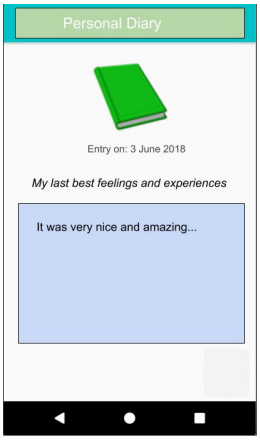
Screen 7

This screen is setting screen. The user can allow sending reminder / notifications.



Screen 8

This screen provides the widget and show the last entry of the journal.



Key Considerations

How will your app handle data persistence?

The Application uses both Content Provider and Firebase Realtime Database. The user will be able to post his daily impressions in real time and see the whole journal. To secure the data the password and the user name will be saved using Firebase auth.

Describe any edge or corner cases in the UX.

On screen 2: Sign in, if the user already have an account or register. If the user pressed “sign out” button, go back to first screen (MainActivity).

Common project requirements.

Generally I am going to use
- **Java** Programming Language for this App.
- Gradle: **`classpath 'com.android.tools.build:gradle:3.0.1'`**

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

- Firebase UI auth for sign in, Firebase storage (for images) and Firebase realtime database:
  `'com.google.firebase:firebase-core:15.0.0'`
  `'com.google.firebase:firebase-database:15.0.0'`
  `'com.google.firebase:firebase-auth:15.0.0'`
  `'com.google.firebase:firebase-storage:15.0.0'`
  `'com.google.firebase:firebase-messaging:15.0.2'`
- Firebase job dispatcher for schedule the job and sending reminder:
  `'com.firebase:firebase-jobdispatcher:0.7.0'`
- Glide to handle the loading and caching of images:
  `'com.github.bumptech.glide:glide:3.6.1'`
- Butterknife to bind the views and annotations: **`'com.jakewharton:butterknife:8.8.1'`**
- Stetho to debug database: **`'com.facebook.stetho:stetho:1.5.0'`**
- Okhttp3 for network: **`'com.squareup.okhttp3:okhttp:3.9.0'`**
- Cardview, Recyclerview for Layout:
  `'com.android.support:recyclerview-v7:27.1.0'`
  `'com.android.support:cardview-v7:27.1.0'`

- Design library: **`'com.android.support:design:27.1.0’`**
- Appcompat: **`'com.android.support:appcompat-v7:27.1.0'`**

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

I will use google play services for authentication and storage of data:
`classpath` **`'com.google.gms:google-services:3.2.0'`**

Next Steps: Required Tasks

Task 1: Project Setup

My subtasks are:
- Create project in Android studio
- Configure libraries
- Create project on the Firebase console

- Create layouts
- Build UI for Activities and fragments
- Testing

Task 2: Implement UI for Each Activity and Fragment

List the subtasks:

- Build UI for MainActivity
- Build UI for LogIn Activity and Register Activity
- Build UI for input data for one day
- Build UI to show all entries and best days list
- BuildUI for Setting screen
- Build UI for App Widget

Task 3: Create model for the data

- Create a model Class for User
- Create a model Class for Entries

Task 4: Create and configure the Content Provider

- Create a Contract Class
- Create a Content Provider Class
- Create a Database Class

Task 5: Configure Firebase rules and classes for storage, database and auth

- Create rules and managing classes for Firebase Realtime Database
- Create rules and managing classes for Firebase Auth
- Create rules and managing classes for Firebase Storage

Task 6: Add Firebase Job dispatcher

- Add library for job dispatcher
- Add a Job Service Class

Task 7: Add settings for color and notifications

- Add a “sandwich” menu for setting color
- Add a setting screen for allow reminder / notifications
- Add logic to use User Preferences

Task 8: Configure Widget and it’s service

- Create Widget Provider

- Create Widget Remote Service

Task 9: Publish the Diary

- Add Signing Configuration in build.gradle
- Publish the App on Github

For Reviewer

I hope that now it is OK and I can start with my Capstone project.

Thanks in advance
Regards
Veronika