GitHub Username: syndicator

Goal Academy

Description

Goal Academy will support you to change your habits for the good! It helps you on your way to achieve your personal goals in terms of including beneficial habits in your life.
A goal might be “Don’t smoke!” or “Eat healthy at least twice a week”.

#1 Set a goal
#2 Track your progress
#3 Be rewarded with cool badges!

Intended User

The app addresses users who want to improve their life through forming useful habits or avoiding bad habits.
There are no constraints regarding the user group.

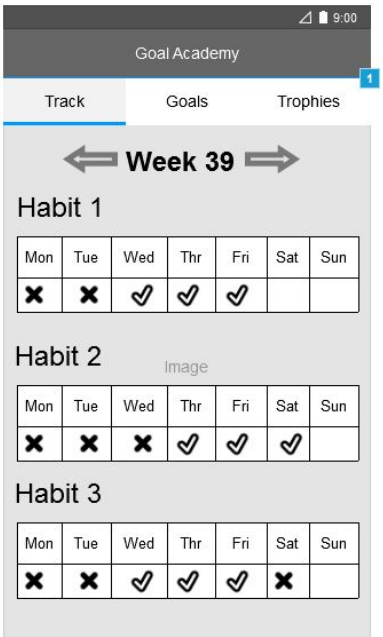
Features

- Provides week-based overview of goals with statistics
- Records and displays on a daily basis if the habits are successful
- Displays a gallery of trophies, which were awarded for successfully accomplishing the plan
- The goal tab will provides motivational quotes pulled from an API
- Widget providing an overview of all habits who are due on this day

User Interface Mocks

The app can basically be operated by three main sections, organized in three tabs.
According to the “Material Design Guidelines on top-level view strategies” I opted for tabs for the following reason:
“Use tabs if:
You expect your app's users to switch views frequently.
You have a limited number of top-level views.
You want the user to be highly aware of the alternate views.”
The tabs are ordered according to importance from left (most important, top level page) to right.

Screen 1: Track Tab



Habit 1

Mon	Tue	Wed	Thr	Fri	Sat	Sun
✗	✗	✓	✓	✓		

Habit 2

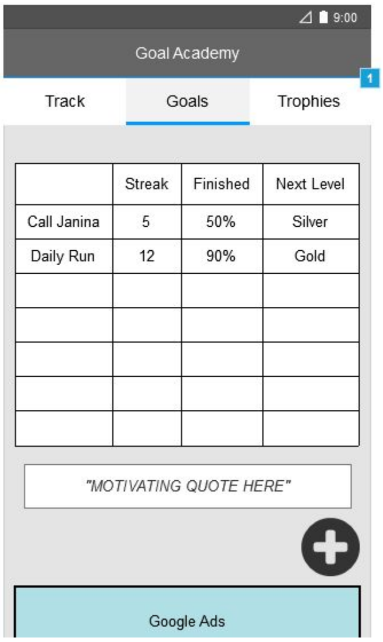
Mon	Tue	Wed	Thr	Fri	Sat	Sun
✗	✗	✗	✓	✓	✓	

Habit 3

Mon	Tue	Wed	Thr	Fri	Sat	Sun
✗	✗	✓	✓	✓	✗	

This is the first and most important tab, as is easily shows the progress in the current week. The user checks and/or unchecks if he achieved the actions on specific days. The week can be scrolled back/forth with the horizontal arrows. Tapping the week will center on the current week. The page shall be scrollable vertically if the habits exceed available space. The tracks will have a color scheme to support the user to know if it is critical to fulfill a habit on this day.

Screen 2: Goals tab

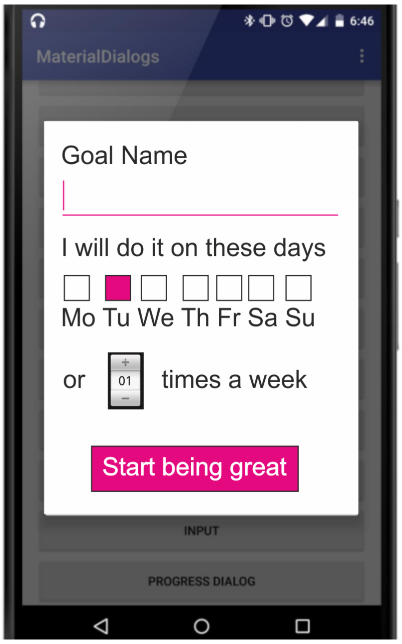


	Streak	Finished	Next Level
Call Janina	5	50%	Silver
Daily Run	12	90%	Gold

This tab gives the user an overview how well they are doing in regularly sticking to a habit. The streak is the number of the incidents successfully passed without missing a critical scheduled incident. Also, the percentage of incidents needed to complete the current level are displayed in the row “Finished”. “Next Level” shows the next trophy to gain.

Below this table, a motivational quote will be shown, which will be updated once a day via an API.
The FAB will allow the user to enter a new goal.
On the bottom, ads are shown in the free version of the app.

Screen 2a: Enter new goal dialog



Goal Name

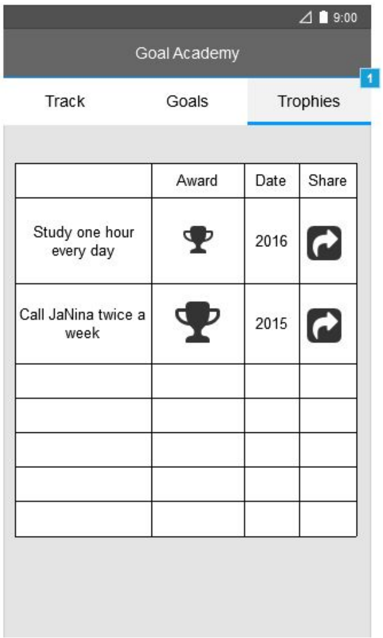
I will do it on these days

□ ■ □ □ □ □ □
Mo Tu We Th Fr Sa Su

or 01 times a week

This screen lets the user enter the goal name and select the days on which they have to do it in order to pass the week to be successful. Alternatively the number of days can be provided if the user doesn't want to do it on specific days. Only one of the two options can be selected at the same time.

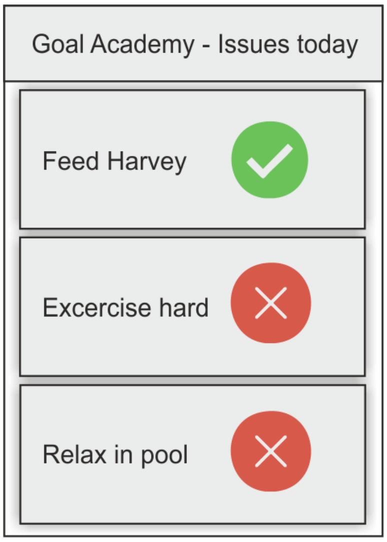
Screen 3: Trophies



	Award	Date	Share
Study one hour every day	🏆	2016	↗
Call JaNina twice a week	🏆	2015	↗

This shows all goals which have been achieved to a bronze trophy at minimum. Available achievement levels (for version one of this app) will be bronze - silver - gold. The number of streaked incidents needed to prevail in one level will be ~ 5 - 21 - 66, based on popular habit research, estimating the minimum habit days at around 21 and the average about 66. A sharing function will allow to share the achievement with an implicit intent to various messaging apps.

Widget



Goal Academy - Issues today

Feed Harvey ✓

Excercise hard ✗

Relax in pool ✗

The widget will display a collection of the goals who have to be worked on today. It is scrollable and a tap opens the app on the starting page (Track).

Key Considerations

How will your app handle data persistence?

Firebase will be used to store the goal, track and trophy data.
For the quotes function, a content provider will be used to store the data and a loader will update the GUI.

Describe any edge or corner cases in the UX.

- Before the user goes online at least once, the “quotes” text will be replaced by a placeholder text
- If no new quotes can be pulled, a random quote from the content provider is used

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

Butterknife - reduce findViewById calls and boilerplate code#
Timber - a logging utility library. Makes logging easier, for example no need to apply tags manually
Gson - gson might be needed for manipulating the data retrieved from Firebase
Retrofit - will pull the quotes data from an API

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

Google Mobile Ads SDK for Android - `com.google.firebase:firebase-ads:11.6.0`
In conjunction with firebase, the above version “AdMob with Firebase” needs to be used. I want to display a banner on the”goals” tab in order not to distract the user too much from app functionality..

Google Firebase Realtime Database
The data will be be persistent and synced over different devices.
The DataSnapshot class can be used to easily process the data.

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

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries
- Something else

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

- Configure project flavours for debug, release, free, paid and corresponding combinations
- Configure gradle dependencies for
    - Butterknife
    - Timber
    - Gson
    - Firebase
    - Admob (Firebase Ads)
    - Retrofit
- Add the google-services.json file

Task 2: Implement UI for Each Activity and Fragment

- Build Main UI (MainActivity) with Tabs and SwipeViews
- Build UI for Track screen (Fragment)
    - Build UI for single track items
- Build UI for Goals screen (Fragment)
    - Build UI for goal items
- Build UI for Trophies (Fragment)
    - Build UI for trophy items
- Build the add new goal dialog UI
- Build the widget UI

Task 3: Implement Google Play Services

- Configure services
- implement Admob ads for free version

Task 4: Implement data persistence

- Implement Firebase Realtime Database into the application to store the data locally in order to show data at any available time regardless of the device's network state
- Implement a Sync Adapter by extending the AbstractThreadedSyncAdapter class
- In onPerformSync, use Retrofit REST client library to get the latest quotes from the API
- Implement a content provider to store the quotes
- Implement a loader to update the GUI

Task 5: Implement UI tests

Create UI tests to automatically test the application UI.

Task 6: Prepare app for release

Test app and fix bugs. Prepare the play store listing with following pre-publish steps:
1. Create app icon
2. Create splash screen for app
3. Prepare store listing