
GitHub Username: NikolayGH

The One Thing

Description

THE ONE THING : THE SURPRISINGLY SIMPLE TRUTH BEHIND EXTRAORDINARY RESULTS. You want more productivity from your work. More income for a better lifestyle.

The application designed to plan your tasks according to the principles outlined in the book "The One Thing" written by authors and real estate entrepreneurs, Gary W. Keller and Jay Papasan. The book discusses the value of simplifying one's workload by focusing on the one most important task in any given project.

"What's the ONE Thing you can do such that by doing it everything else will be easier or unnecessary?"

The widget on the screen will constantly remind you of the most important goals for the nearest stretches of time, thereby constantly directing the user's attention to key tasks during daily routine.

Intended User

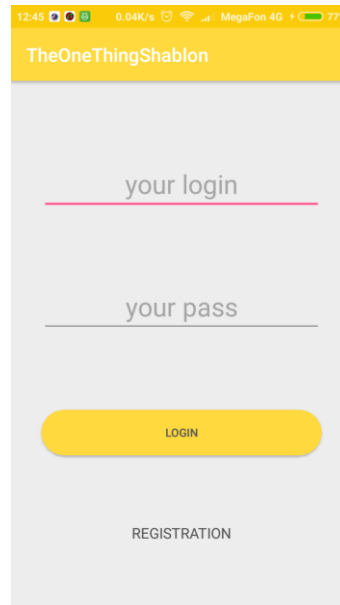
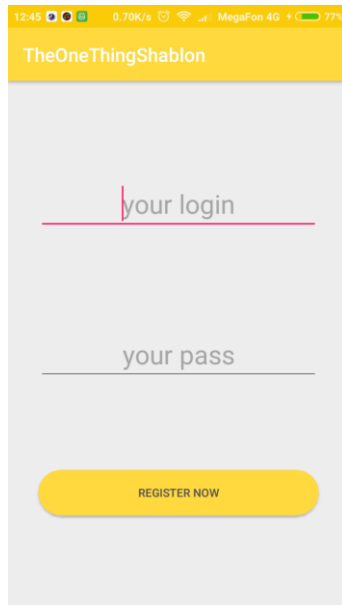
The application is intended for a wide range of users: from managers to housewives. For all those who want to achieve their goals.

Features

- Creates the most important task for current day, week, month and year.
- Adds description for task
- Takes pictures of result for completed tasks.
- Saves information
- Sets widget on home screen with your tasks for day, week, month, year.
- Shows your past tasks

User Interface Mocks

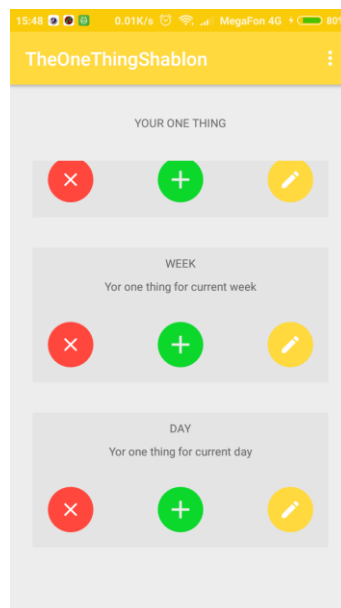
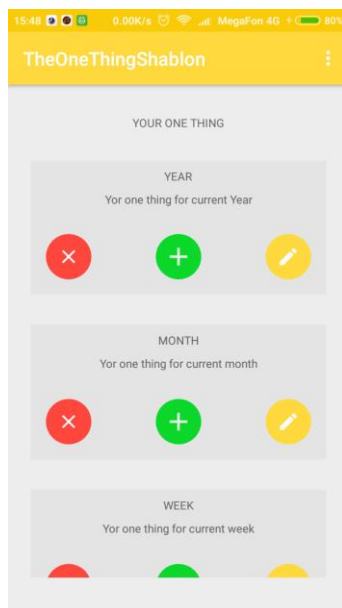
Screen 1



or

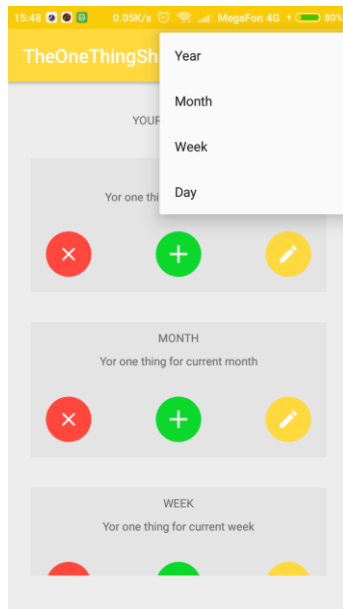
On first screen user will do registration or login (if user is already registered).

Screen 2



by scrolling:

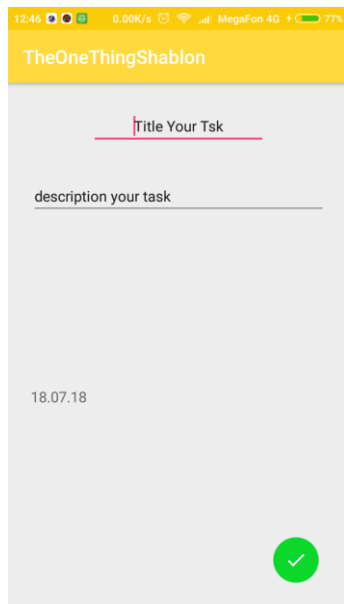
On second screen user can see, edit and add only his most important task on any future time period ("The One Thing" for each time period). By clicking on edit button will opened screen 3.



by menu clicking:

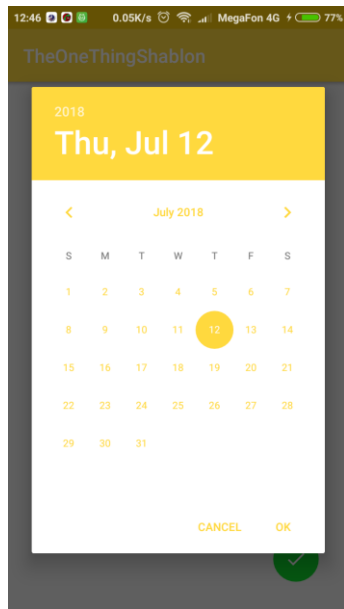
By on item menu clicking user can see screen 4 with tasks from selected time period

Screen 3

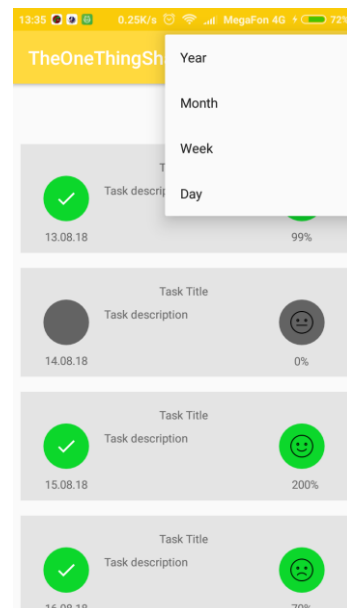
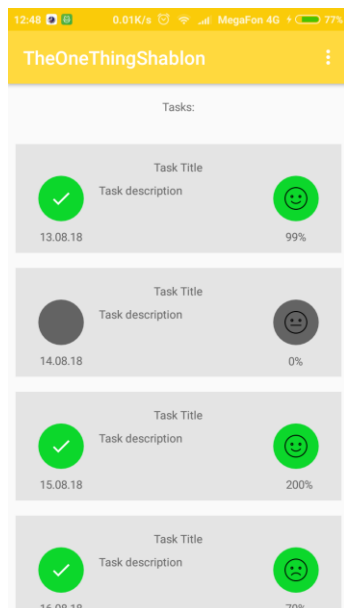


Here user will set title and description of his task

by clicking on date user can set any date for your task:



Screen 4

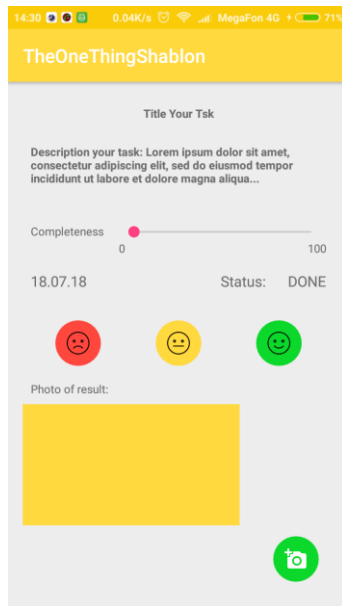


By menu user can select type of tasks:

Left button in item shows status of task (complete or not complete). By clicking on it user can change the status. Right button shows emotional assessment. Percents bottom of button shows completeness of task.

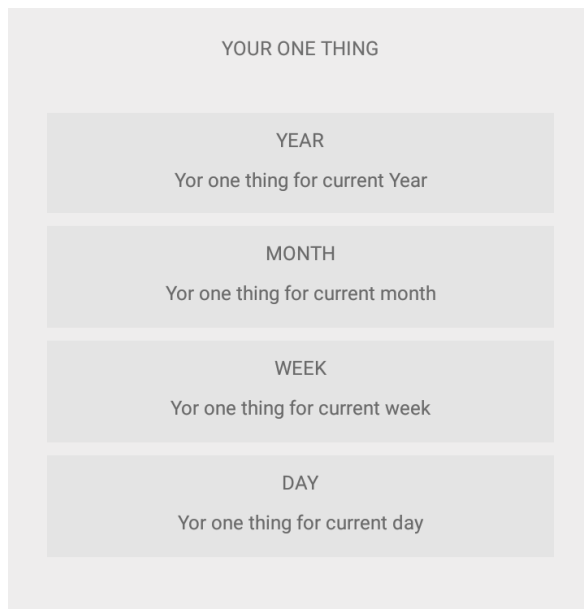
By item menu clicking user can change time period for tasks. By clicking on any task will open screen 5

Screen 5



Here user can set degree of completion of the task, take a photo of result and set assessment of his emotions by working on task.

Widget



Widget for home screen contains task from all time periods to remain priority among routine tasks for user.

Key Considerations

How will your app handle data persistence?

I will use firebase for to:

Authorization

Authentication

To store data on server in firebase database

I will use Realm to:

Store data on device (additional)

Describe any edge or corner cases in the UX.

Synchronization widget information with Firebase or local database

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

Realm (additional)

RxJava

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

Firebase, Cloud Storage

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

- Configure libraries Realm, RxJava
- Update Google Play services
- Create Project in Firebase console
- Configure Firebase Authentication

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainTasksActivity
- Build UI for RegistrationActivity
- Build UI for LoginActivity
- Build UI for EditTaskFragment
- Build UI for TaskCompleteFragment

- Build UI for TaskListFragment
- Build UI for DatePickerDialog

Task 3: Implement authorization and authentication (screen 1)

- Implement authorization via Firebase
- Implement authentication via Firebase

Task 4: Implement Firebase database

- Implement Firebase database
- Connect to database

Task 5: Implement adding and deleting tasks (Screen2, 3)

- Create recyclerView for tasks
- Add listener for data in database
- Transfer data via adapter
- Implement adding (screen3) and deleting tasks
- Implement menu for selecting tasks by time period
- Set filter to selecting data for main tasks list (screen 2)

Task 6: Implement sort and show tasks by time period (Screen 4)

- Create recyclerView for tasks
- Show selected tasks in recyclerView
- Implement checking for tasks

Task 7: Implement editing completed and uncompleted tasks (Screen 5)

- Implement setting degree of completion of the task
- Implement emotional degree
- Implement Cloud Storage
- Implement adding photo to tasks

Task 8: Create widget

- Create appwidget-provider
- Create widget data model
- Create widget service

Task 8: Implement database for local storage (additional)

- Create data models
 - Create class for migration
 - Create DB Service
 - Implement showing tasks without internet connection
-