

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement Firebase Database](#)

[Task 4: Implement Sign Up and Login using Firebase Authentication](#)

[Task 5: Implement Task Lists](#)

[Task 6: Implement Tasks](#)

[Task 7: Sync Database](#)

[Task 8: Settings Activity](#)

[Task 9: Bughunt](#)

GitHub Username: Mika412

OpenDo

Description

OpenDo is a simple, open source smart task list that makes it easy to plan your day. Whether for work, school or home, OpenDo helps you increase productivity and reduce stress.

Intended User

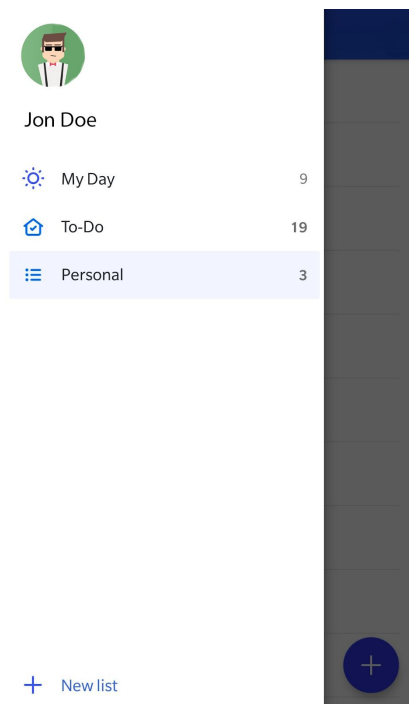
This app is for the people who want to get their life on track and be more productive, by planning their days with a to-do list.

Features

- Saves information locally and on the cloud
- Syncs between devices
- Add new tasks
- Create new lists
- Add description to the task
- Add a reminder for the task
- Widget that displays the desired tasks on the home screen

User Interface Mocks

Screen 1



Side menu that lists all Todo Lists, with an option to add a new list.

Screen 2

Personal


- ☐ Workout
- ☐ Go shopping
- ☐ Pick up Grandmother

+

List of the lists todo items, with a button to add a new item.

Screen 3

← Settings

 Jon Doe

SIGN OUT

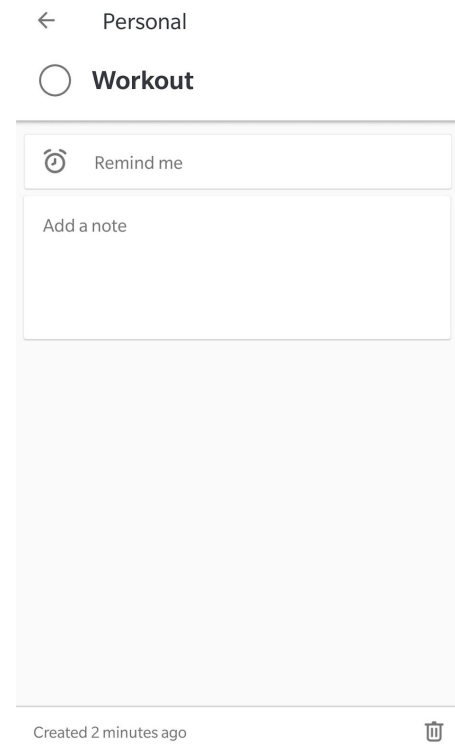
General

Theme Blue

Display reminder notification
Display notification for Todos with a reminder ☒

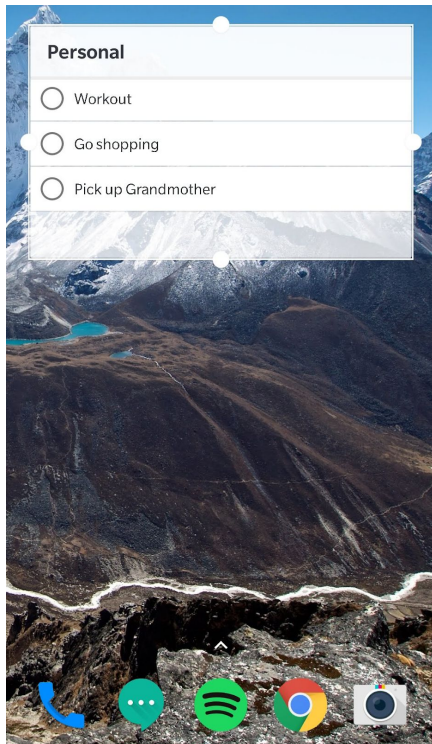
Settings screen, where you can logout, change theme or disable notifications.

Screen 4



Todo description. By clicking a todo item, this is the screen that appears with all the details about the item and with the ability to change them.

Screen 5



Widget that lists the desired List tasks/ToDo's.

Key Considerations

How will your app handle data persistence?

It uses Firebase Realtime Database to backup and sync data between devices.

If users data is changed in the database, the server(Firebase cloud function) will send an FCM to all users devices. Then the clients will start a background service that will sync the users data with the database.

Describe any edge or corner cases in the UX.

If the task is opened, the user can click on the back button to go the previous list. To enter settings, the user has to click on his profile picture.

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

Firebase library to import the necessary Firebase services and Butter Knife to automatically cast the correct views from the layout.

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

It will use Firebase Authentication as the login system and the Firebase Database for the database sync.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Create an app project on Firebase
- Insert the Firebase key into the Android project.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for ListTasks Fragments
- Build UI for ListLists Fragment
- Build UI for SettingsActivity

Task 3: Implement Firebase Database

- Design database layout
- Implement database

Task 4: Implement Sign Up and Login using Firebase Authentication

- Implement Sign Up function
- Implement Login
- Implement Forgot Password / Reset function

Task 5: Implement Task Lists

- Create Display Lists
- Add new List
- Sync list with database

Task 6: Implement Tasks

- Create new task function
- Change task list
- Add a task description
- Add a task alarm
- Sync list with database

Task 7: Sync Database

- On login, sync app with database
- Create a background service that syncs with database
- On receiving an FCM, start the background service

Task 8: Settings Activity

- Implement Logout
- Implement Delete Account
- Implement Change App Theme

Task 9: Widget

- Design widgets UI layout
- Implement Widgets creation and resizing
- Make the widget display the Lists ToDo tasks
- Make the widget update the displayed Todos, if any change happens

Task 10: Bughunt

- Search for bugs
- Solve bugs