**GitHub Username**: @shahidcodes

# Turn Reminder

## Description

Basically when some people share a flat, there are various task assigned to each of them. For example- Cleaning, Dish Washing, Cooking ETC. Sometime it is like one person have to do something every 4th sunday if there are 4 people. So it's easy to forget who did what last last Sunday. However this is managed by writing each person's name on paper and making checkboxes etc. Right now me and my flatmates are facing this issue to forget what they have to do this weekend or next weekend. So this app will let user add task and get reminded of their task before the weekend.

## Intended User

All the Students/Bachelors sharing a room or flat.

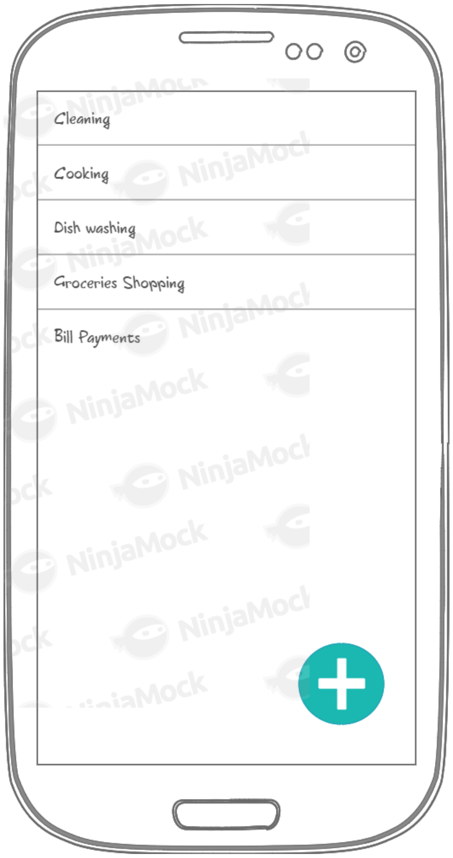
## Features

List the main features of your app. For example:

- Adds a task.
- Reminds to finish the added task.

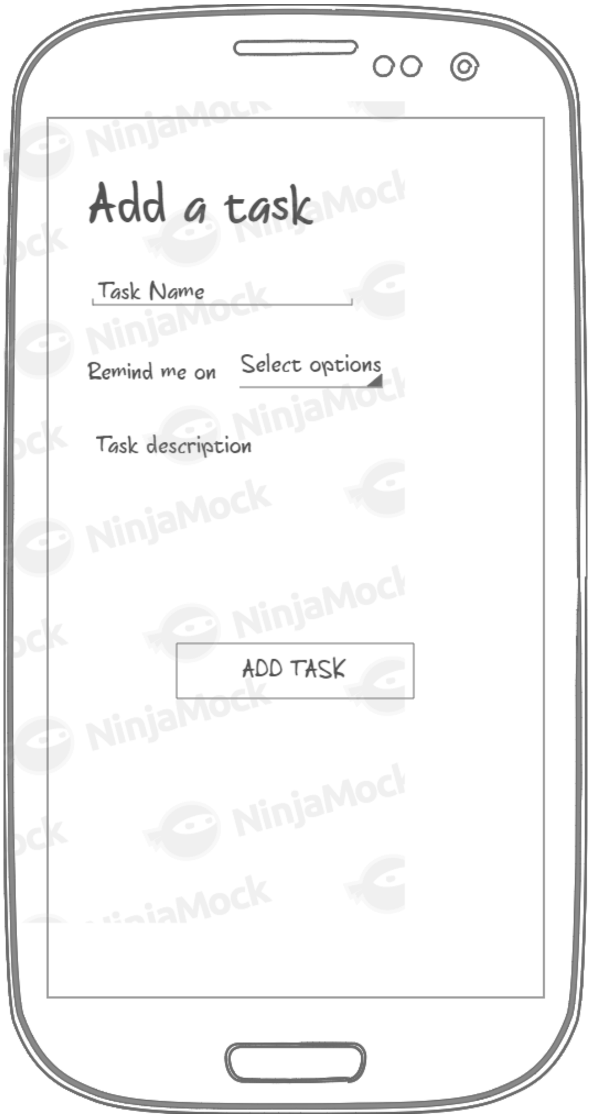
## User Interface Mocks

### Screen 1



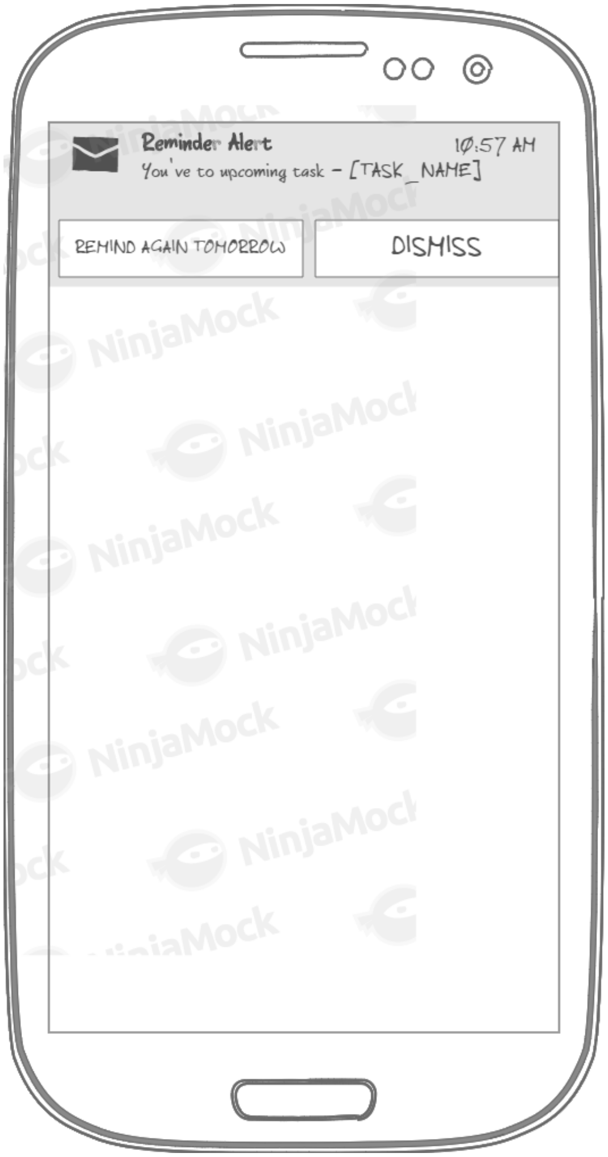
Screen will be presented when user launches the app. If it's first time they will have some pre populated data. In this screen they can click on task which will open the next screen with populated task details

## Screen 2



This will open up when use clicks on FAB on main screen. This will be reused to show the existing task to update or delete

# Notification View



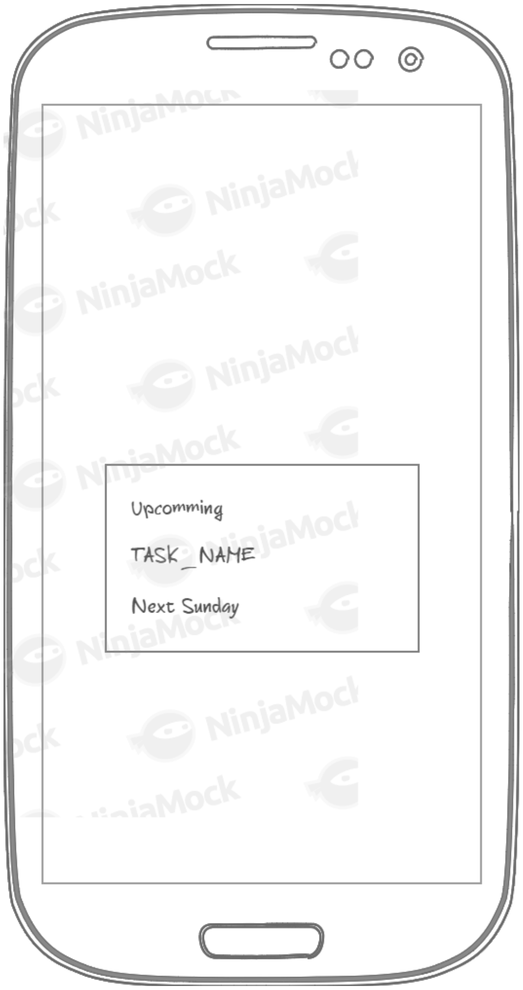
Reminder Alert 10:57 AM

You've to upcoming task - [TASK_NAME]

Example reminder notification when user have upcoming task

## Widget UI:

Widget will contain the information about next/upcoming task that user has to finish. Clicking on it will open an activity to show the task information.



# Key Considerations

### How will your app handle data persistence?

I'll use Room from Android Jetpack to store data locally.

### Describe any edge or corner cases in the UX.

For tablet screen or larger there'll be master-details flow layout to show tasks and task details side by side.

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

Room from Android Jetpack- It'll make implementation easier.

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

Google play services will be used to implement Admob and Firebase Crash.

# Technical Tasks

### App integrates a third-party library..

Butterknife will be used to bind views.

### App integrates two or more Google services.

Admob and Firebase Crash Analytics will be used.

### How app will handle pulling data ?

Room library with Async Task. Data will be loaded on launch to show list of tasks from db.

### RTL and Accessibility Support

- Application will support RTL.
- All values i.e. texts, color values, dim etc will be kept in values folder in their respective files.
- Content description will be provided for every component.

### Project Setup

**Language:** Application will be written solely in the Java Programming Language.

**Android Studio: 3.2.1 Stable Version**

**Gradle:** Latest and greatest. At time of writing it is Gradle 4.6

# Next Steps: Required Tasks

## Task 1: Project Setup

- Create project fix all the target sdk versions, add google() in repository and other places if it’s not available
- After sync is completed, add the Room, Android Support Design, RecyclerView etc libraries
- Library versions:
    - Room 2.x
    - Design Support Library 27.x
    - RecyclerView 27.x
    - Butterknife 8.x
    - Firebase Admob 17.x
    - Firebase Crash Analytics 16.x
    - Other libraries will used in their latest stable version if need arose during project development

## Task 2: Implement UI for Each Activity and Fragment

- Create MainActivity and a MainListFragment for listing all tasks from db
- Create a TaskActivity and TaskFragment for editing and adding new tasks.
- Create Layouts for tablet and show MainListFragment and TaskFragment in Master Details Flow
- Create boolean value for tablet layout to easily determine if its tablet or phone.
- Implement cases for click on list items when on tablet to replace the TaskFragment with new task item. And open task activity if it is on phone.

## Task 3: Write Tests

- Write the test if RecyclerView is populated
- See if clicking on item works normally on phone and tablet
- Write test for creation of new task.
- Write test for notification

## Task 4: Build Variants

- Build free and paid variants
- Show ads in free variants
- Remove unnecessary libraries/codes from paid variant

## Task 5: Write gradle task for building and running tests

- Write a combined task to build and run all tests