

How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Key Considerations](#)

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

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Set Up Data Handling](#)

[Task 4: Implementing Google Play Services](#)

[Task 5: Creating AppWidget](#)

GitHub Username: [VamsiSmart](#)

Tenants Manager

Description

“Tenants Manager” is very useful for managing small scale, medium apartments and private hostels.

it has two main features, Tenants and Notes.

Owners can add new tenants and manage tenants, they can see the all tenants contact details rent dues. They can note down complaints set reminders.

Intended User

Medium and small scale Apartment and Private hostel owners.

Features

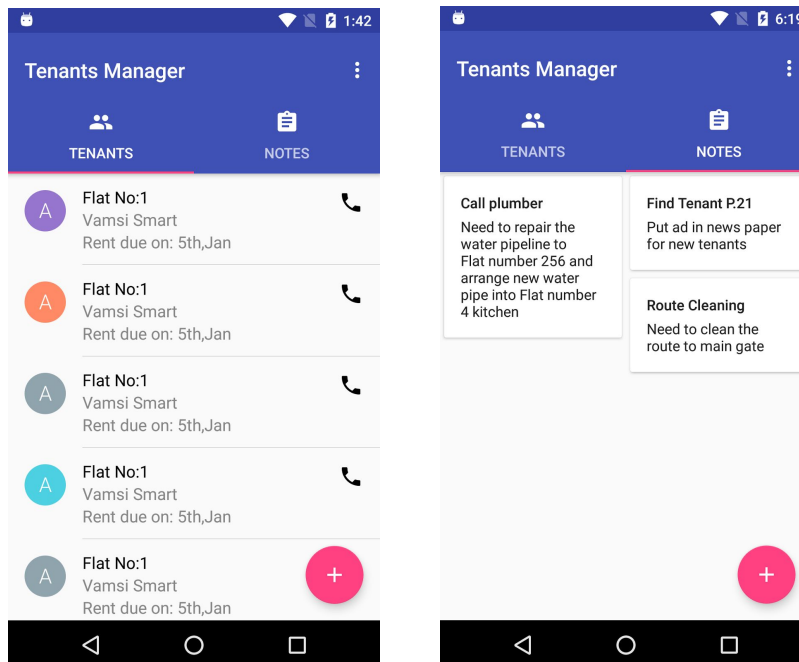
Main features of your app:

- Add tenant
- Manage tenants
- Add notes
- Set reminders
- Backup (to server) and restore (from server) tenants
- AppWidget shows rent due tenants

User Interface Mocks

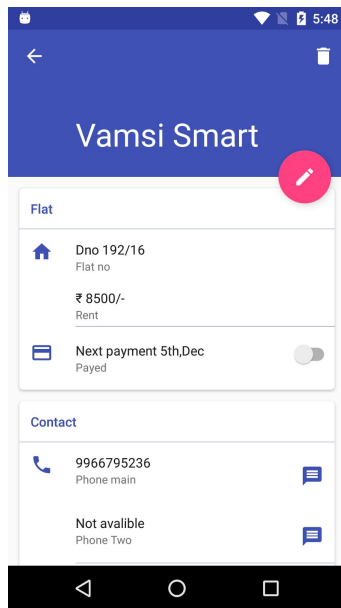
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1 : Home page



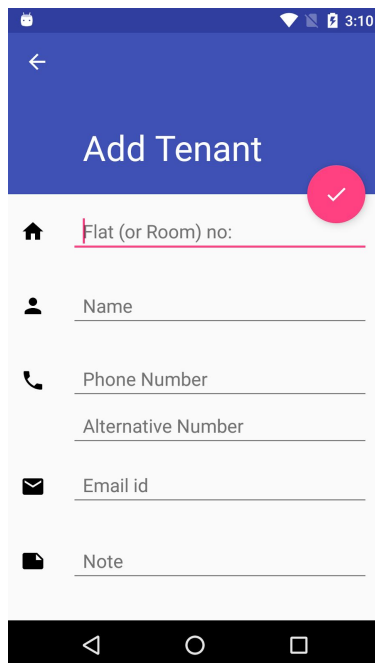
The home Activity is a viewpager it contains two screens Tenants and Notes
First screen display all the list of Tenants, and second screen displays the list of notes.

Screen 2 : Tenant Details



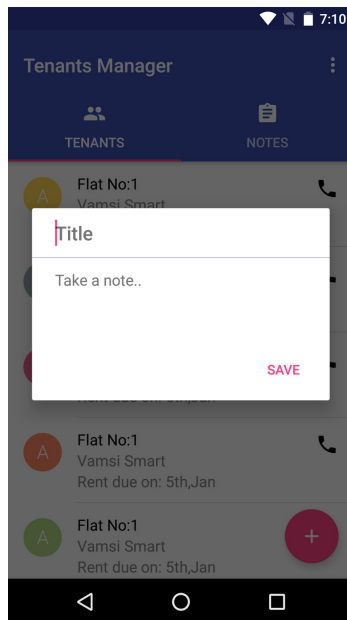
This screen display all the details of selected tenant details including contact details next due date for rent payment

Screen 3 Add tenant



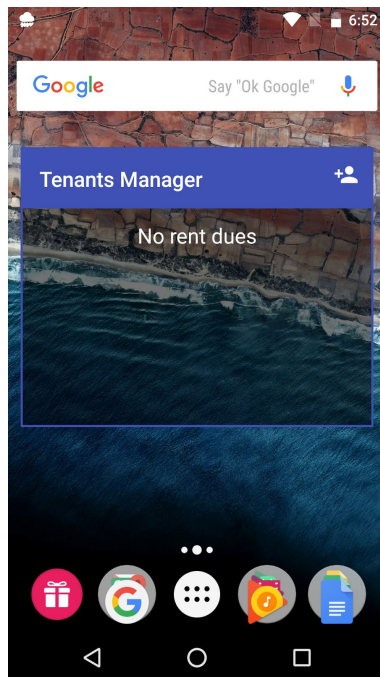
It has few fields to add new tenants,example:Flat number,Name,Rent pay date,contact details.

Screen 4 Create Notes:



Can create a new notes by just entering the title and content.

Screen 5 Widget:



Shows the list of rent due tenants and allow to add new tenant.

Key Considerations

How will your app handle data persistence?

SQLite - for storing data locally .

Content Provide - for CRUD operation.

Loaders - to update UI.

IntentService - for API calls to backup (to server) and restore (from server) tenants

Describe any corner cases in the UX.

The user can open directly the Screen2 when clicked on a reminder notification on the notification tray.

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

TextDrawable - for creating avatars like Gmail App.

Android Asynchronous Http Client for network calls to restore and backup data.

This library is very simple to use

Describe how you will implement Google Play Services.

AdMob: for displaying ads in Screen2 and Screen3

Firebase Analytics to provide app analytics.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Create Simple Project with Tabbed Activity

Change Launcher activity as per app requirements and rename the tab titles as TENANTS and TASKS

Create fragments for each tab , create two Activities as TenantDetails and AddTenant.

Task 2: Implement UI for Each Activity and Fragment

UI for home screen

- Build UI for TenantsFragment
- Build UI for TaskFragment
- Build UI for AddTask Dialogue

UI for Other screens

- Build UI for TenantDetails
- Build UI for AddTenant

Task 3: Set Up Data Handling

Creating content provider, SQLite helper class implementing loaders as per app requirements
Creating IntentService to backup and restore data

Task 4: Implementing Google Play Services

List of subtasks:

- Implementing admob and displaying ads in screen2 and Screen3
- Implementing Firebase Analytics.

Task 5: Creating AppWidget

List of subtasks:

- Setting up AppWidgetProvider class
- Creating a widget for showing number of rent dues and button for opening Add tenant screen

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"