How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

□

GitHub Username: WonkyDan

Unit licence organiser

Description

This app will assist anyone in the UK that is involved in the inspection and licencing of military explosive sites.

It will hold unit and licence details and will send reminders when a location needs an inspection or a new licence.

Intended User

Anyone involved in the task above.

Features

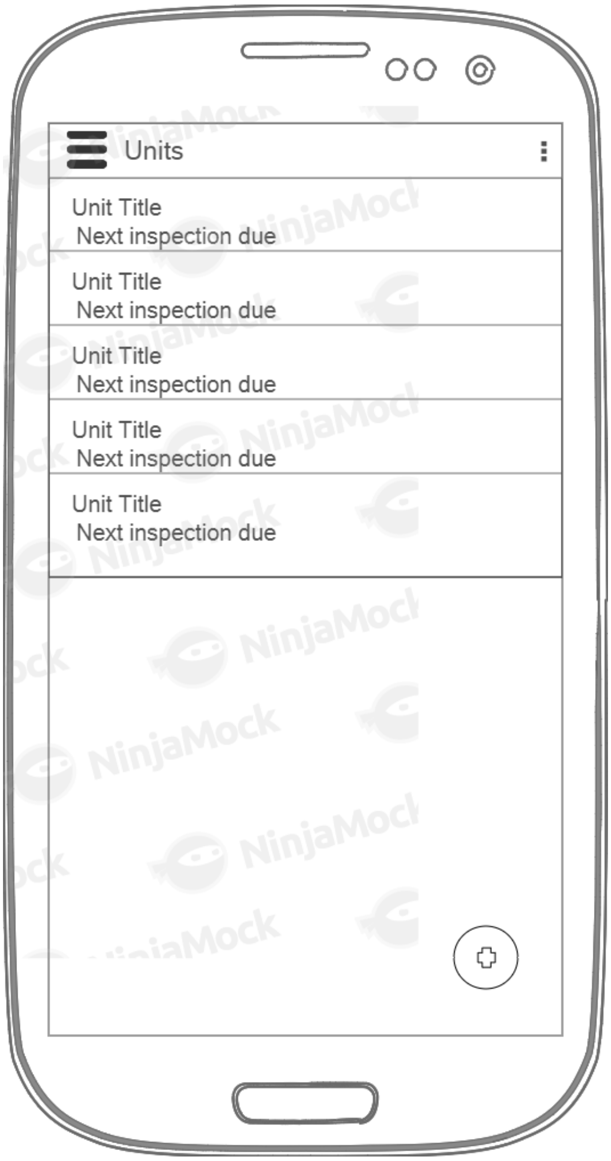
The app will be written solely in the Java language and all libraries, services, Gradle and AS will use only stable release versions.

- An ability to add new units
- A QD calculator to aid in licence creation

- Notifications when licences and inspections are due
- Details of points on previous inspections and reminders to chase up.
- Details of licences held by units.

Add as many screens as you need to portray your app’s UI flow.

Main page showing a list of units. The plus button at the bottom for adding a unit. The menu button top left allows access to the QD calculator via a sidebar.



Units

Unit Title
Next inspection due

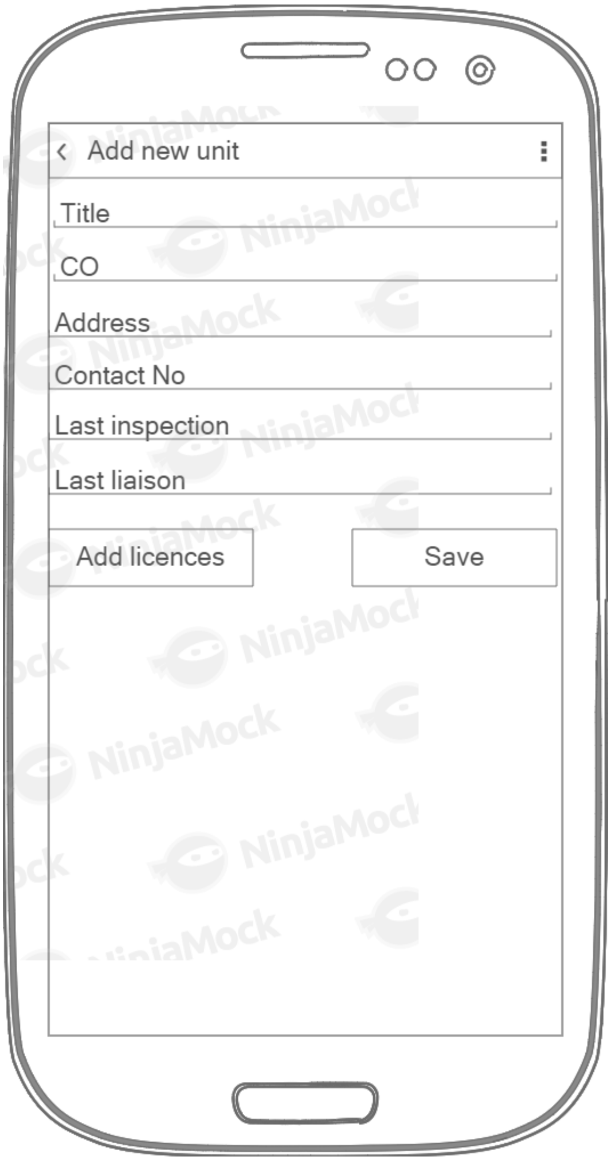
Unit Title
Next inspection due

Unit Title
Next inspection due

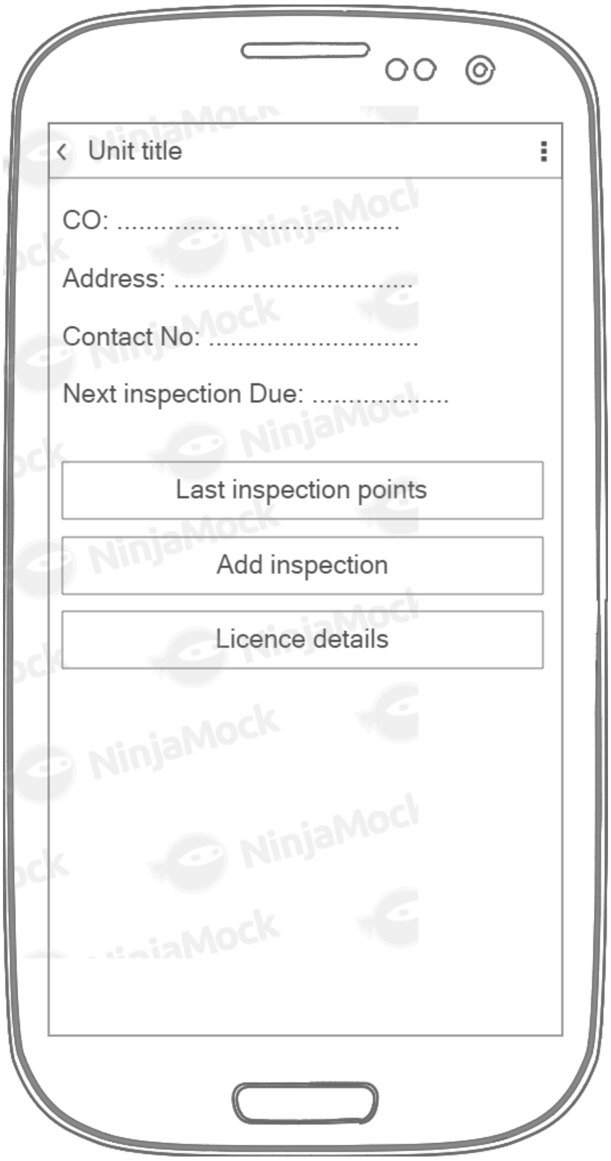
Unit Title
Next inspection due

Unit Title
Next inspection due

The add a unit screen. Licence details are added via a pop up window after clicking the add licences button.



The unit detail screen. The last inspection points can be looked at via the button and each has a check box to show completed or not. A new inspection can be added which will replace the previous points, licence details can also be viewed from this screen.



Unit title

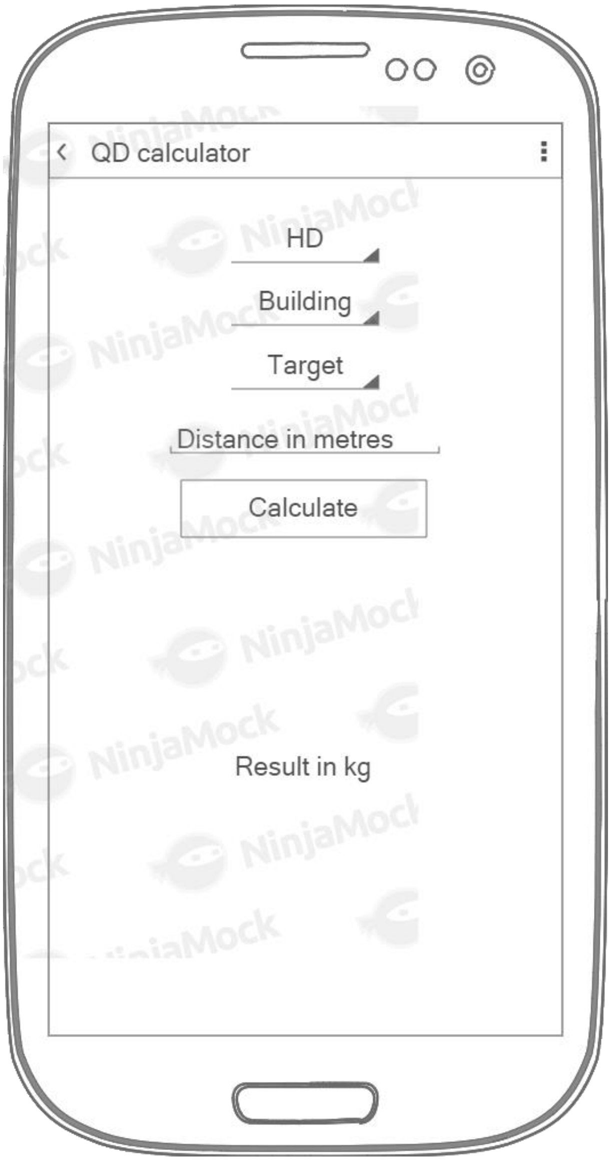
CO: ....

Address: ....

Contact No: ....

Next inspection Due: ....

The QD calculator will calculate the max quantity of explosives that can be stored in a building with the current distances to other locations and type of explosives held.



Key Considerations

How will your app handle data persistence?

I will use room and firebase database to handle the persistent data. I will also need to use GSON to JSON in order to keep the number of databases down.

Room livedata will be used in the unit detail page, ensuring it is kept up to date.

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

I will be using the butterknife library to handle the UI elements as it helps keep the code clean.

UX corner cases.

I will ensure that both left to right and right to left screen layouts will work.

The room database will hold everything required to complete the UI in case of no network signal for accessing firebase.

A master/detail view will be used between the list of units and unit details pages on tablet devices.

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

Firebase database will be used alongside room as a backup.

Firebase analytics will also be used to help identify any areas I could improve in the future.

I will also include Firebase auth to give access to the information across platforms.

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

Add the latest version of the various libraries and dependencies required.
All string resources and themes will be setup.

Task 2: Implement UI for Each Activity and Fragment

- Build all the UI to match the mockups.

Task 3: Implement room

- Create room database
- Create task objects
- Create the view model
- Ensure data completes the UI
- Create objects for licences and inspection points
- Convert those to JSON using GSON to allow entry into the room database.
- Livedata will be used to ensure the UI is kept up to date.
- Ensure that the queries ato the database are done as an Async task.

Task 4: Setup the notifications

- Create notification service
- Create notifications for 28 days after an inspection to chase up any points.
- Create notifications for 2 months prior to a new licence being required or an inspection required.

Task 5: Implement firebase database along with firebase auth.

- Ensure the database gets updated each time the room database updates.
- If this isn't possible the service will be used to update firebase when network becomes available through a jobdispatcher.
- Setup the database so that once logged in firebase is checked using an intentservice and if the information within is newer it will update the room database, if not the room database will be used to fill the UI.
- The firebase database will update in line with the room database using an intentservice.

Task 6: Implement firebase analytics

- Log which parts of the app get used the most allowing me to tweak the app.

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

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
  - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"