Description
Intended User
Features
User Interface Mocks
Screen 1
Screen 2
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: Your Next Task
Task 4: Your Next Task
Task 5: Your Next Task

Java language will be used for development

GitHub Username: **Mahmoud-Heshmat**

Car Assistance

Description

This app will send you notifications for all the things related to your cars, so you don't have to worry about them! The app can be configured to remind you of :
-Insurance expiration
-Vehicle inspection
-Road tax
-Oil change
-Engine air filter
-Cabin air filter
-Battery
-Tire rotation
-Windshield wiper
And many more that car owner add his/her custom reminder

Intended User

The application is aimed to car owners who want to care about their cars and their who forget the time of maintenance

Features

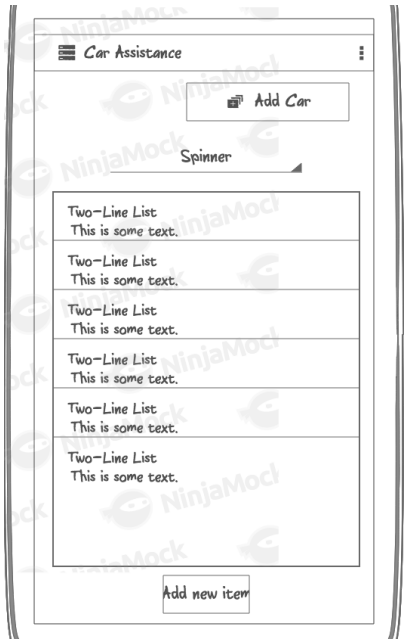
List the main features of your app. For example

- Add car information to make profile
- Add more than one car
- Take notes
- Login by (Username, Password) or Google
- Reminder as (DateTime, Mile, KM)
- Track services and monitor your vehicle maintenance log.
- Multiple vehicles supported

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 Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1

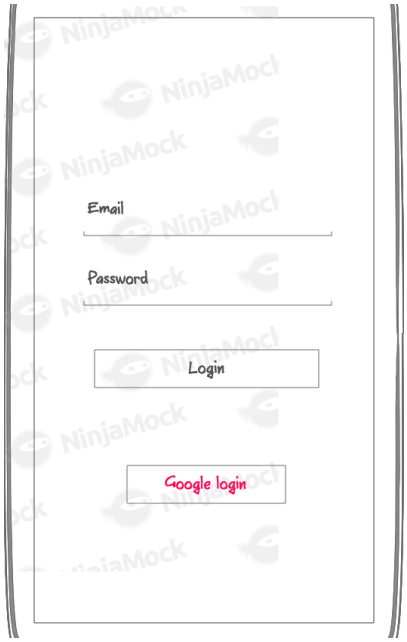


Home screen that have spinner to choose from his/her car and list of car remiderns details
And Sandwich icon for navigation, With two buttons (add new car or add new item)

Screen 2

This screen for add new car with many option and create customize option

Screen 3

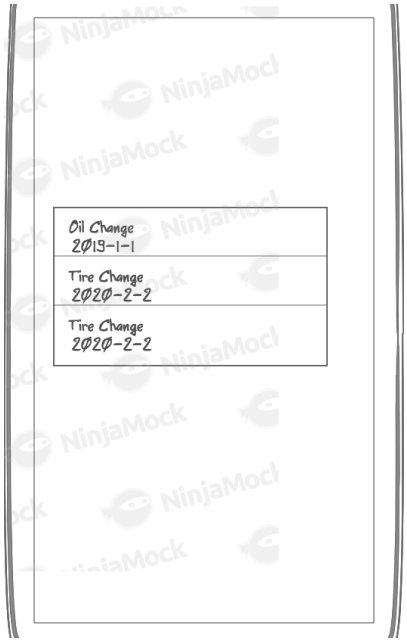


This screen for make login with (email or password) or with Google Sign in

Screen 4

This screen for make sign up or sign using Google API

Screen 5



This screen is widget that display the reminders of all changes the car need

Key Considerations

How will your app handle data persistence?

Describe how your app with handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)

- Shared Preference
- Roon Presentence library
- Connect to back-end services

Describe any edge or corner cases in the UX.

The user will navigate between the screens by clicking on the item on recyclerview or on a button, each detail screen will have up button that navigate the user to previous screen, back button will handled in similar way

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

- Android support library 26.1.0: Handle all UI elements that are supported by the android library, Such as recyclerview, card view, etc.
- Constraint layout support library 1.0.2 : to design layout
- Android architecture component 1.1.1 : Life cycle including LiveData and viewModel
- Volley 1.0.19 : Handle networking call
- ButterKnife 8.5.1: Field and methods binding for android views which use annotation processing to generate boilerplate code
- Firebase 11.8.0: for analytics and crash reports
- Room 1.1.1: Persist data
- Google play service auth 11.8.0 : to make login

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

- Google analytics for analytics and crash reports
- Google auth: Make login with google

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

You may want to list the subtasks. Such as:

- Configure libraries that will be use in the app in gradle
- Add Permissions in manifest
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity that display the list of cars reminders
- App theme will extend appCompat
- App will use App Bar and associated toolbar
- Build UI for car details
- Build UI for add new car
- Build UI for user profile
- Build UI for setting page
- Build UI for notes

Task 3: Handle Error cases

Describe the next task. List the subtasks. For example:

- Handle internet connection access
- Provide placeholder for no image cases

Task 4: Implement Google Play Services

Describe the next task. List the subtasks. Such as:

- Google auth for user login by google
- Google analytics for crash reports

Task 4: Implement Room presentence library

Describe the next task. List the subtasks. Such as:

- Create an Entity model class that will represent a table within the database
- Create a database class by extending RoomDatabase that will contain the database
- Create the DAO interface that will contain the methods used for accessing the database

Task 4: Implement the android architecture component

Describe the next task. List the subtasks. Such as:

- Create ViewModel that will store and manage UI related data in lifeCycle Conscious way
- Use liveData for observing mutable data throughout the app

Task 5: Volley Networking library

Describe the next task. List the subtasks. Such as:

- Create API manager that handle all api request
- Create remote service interface that will hold all the methods

Task 5: Implement widget

Describe the next task. List the subtasks. Such as:

- Create Remote adapter
- Create a widgetProvider

Task 6: Implement JobDispacter

Describe the next task. List the subtasks. Such as:

- app updates data in its cache at regular intervals using a JobDispacter

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**”