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: MarcinWgn

Fuel Bills

Description

The application is used to monitor fuel consumption, fuel price and mileage over time. It helps to optimize fuel consumption and to regulate fuel expenses.

Intended User

The application is intended for car users.

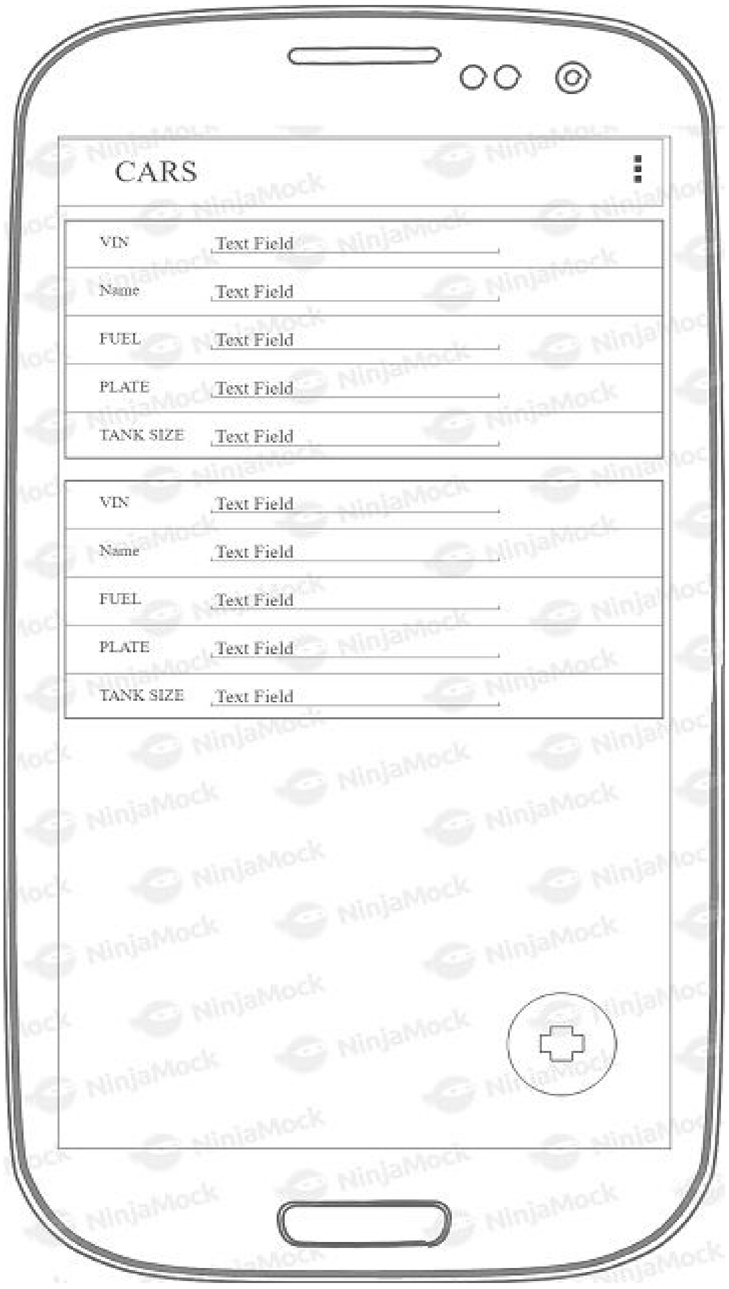
Features

Main features

- Support for many cars
- Fuel consumption calculator
- Charts of an average fuel consumption, fuel price, mileage
- Data synchronization using Firebase Cloud

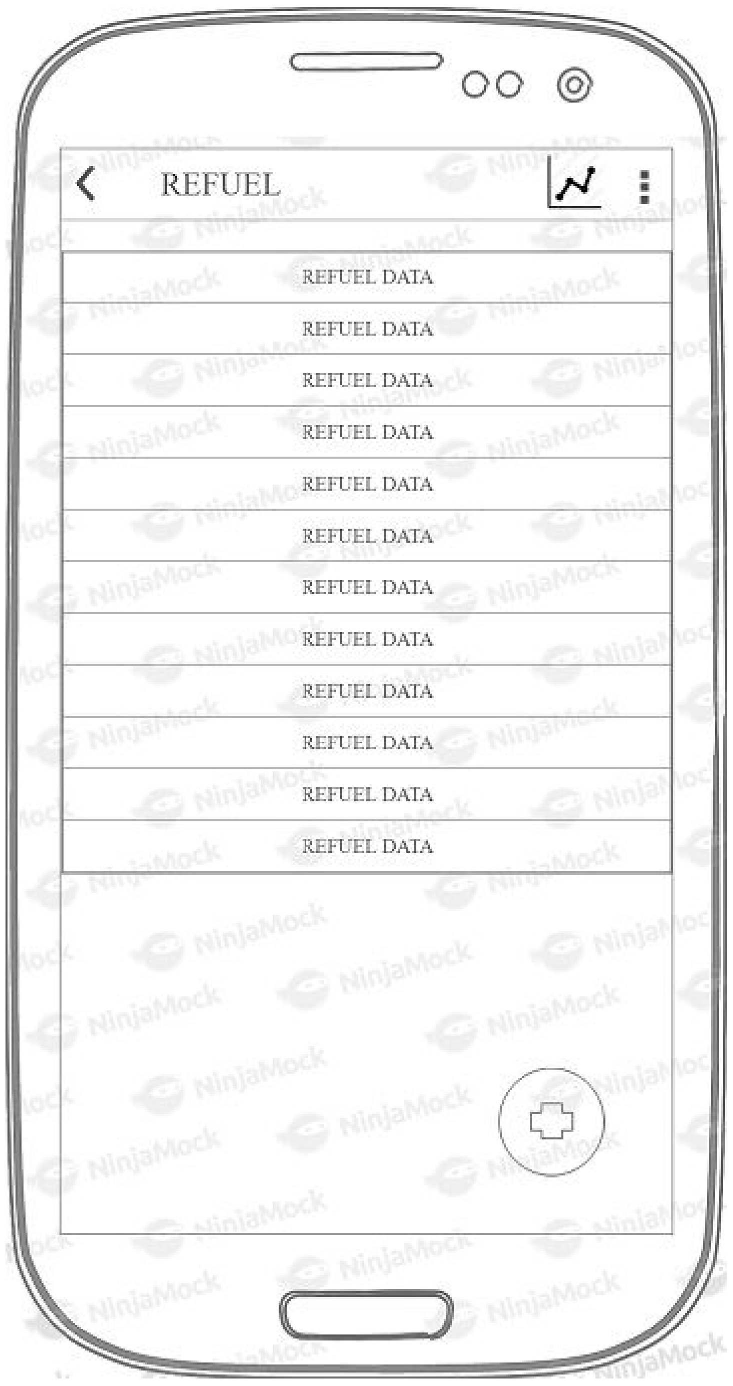
User Interface Mocks

Screen 1



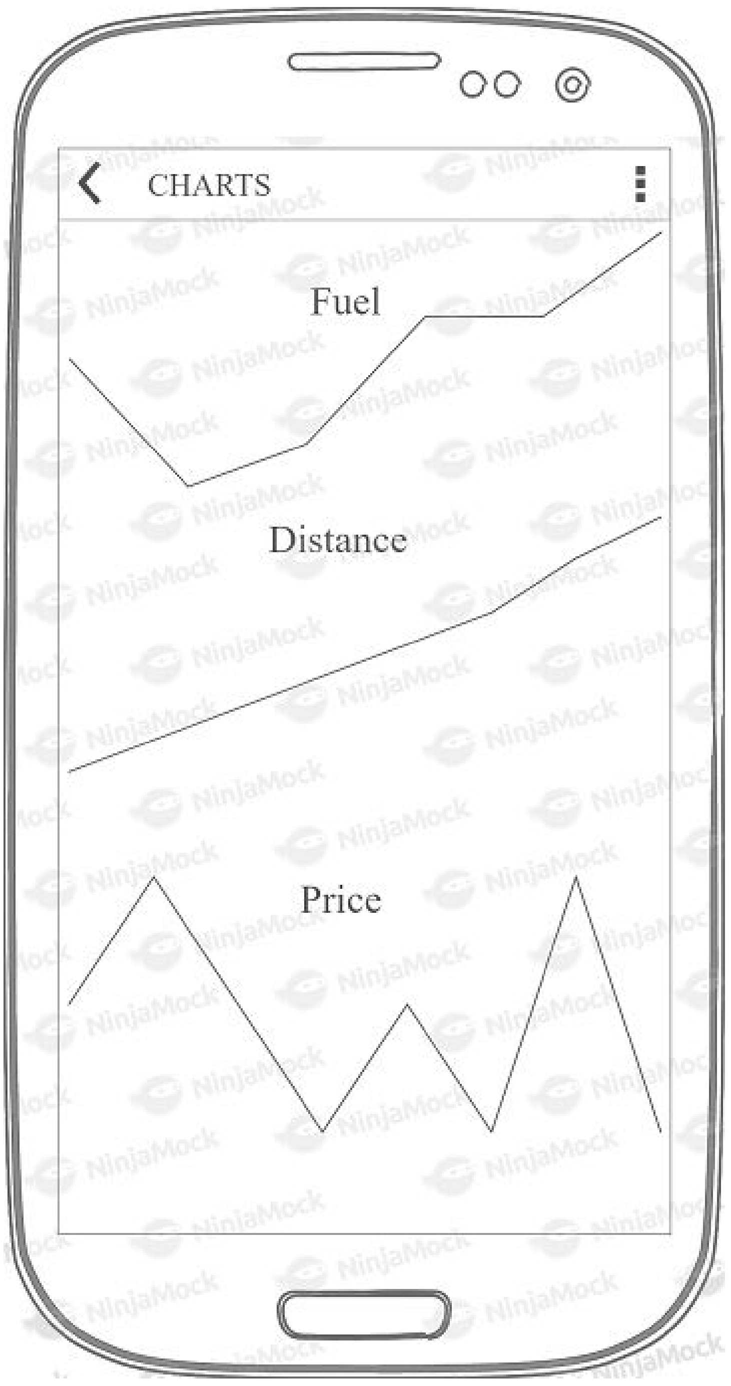
This is the main application screen, displays a list of car cards, allows you to add a new card and navigate to the list of fuel consumption.

Screen 2



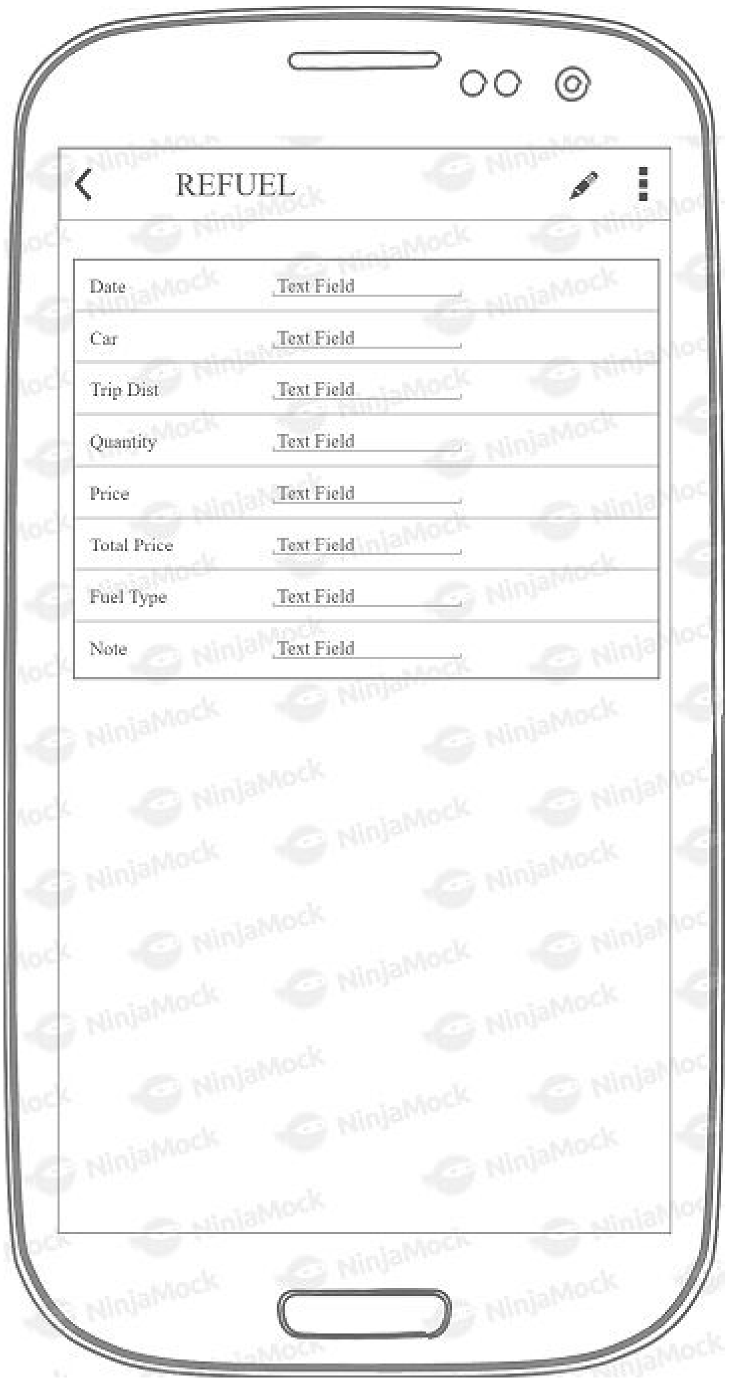
This is the refueling list screen, allows you to add a new refueling and navigate to the charts.

Screen 3



This is the chart screen.

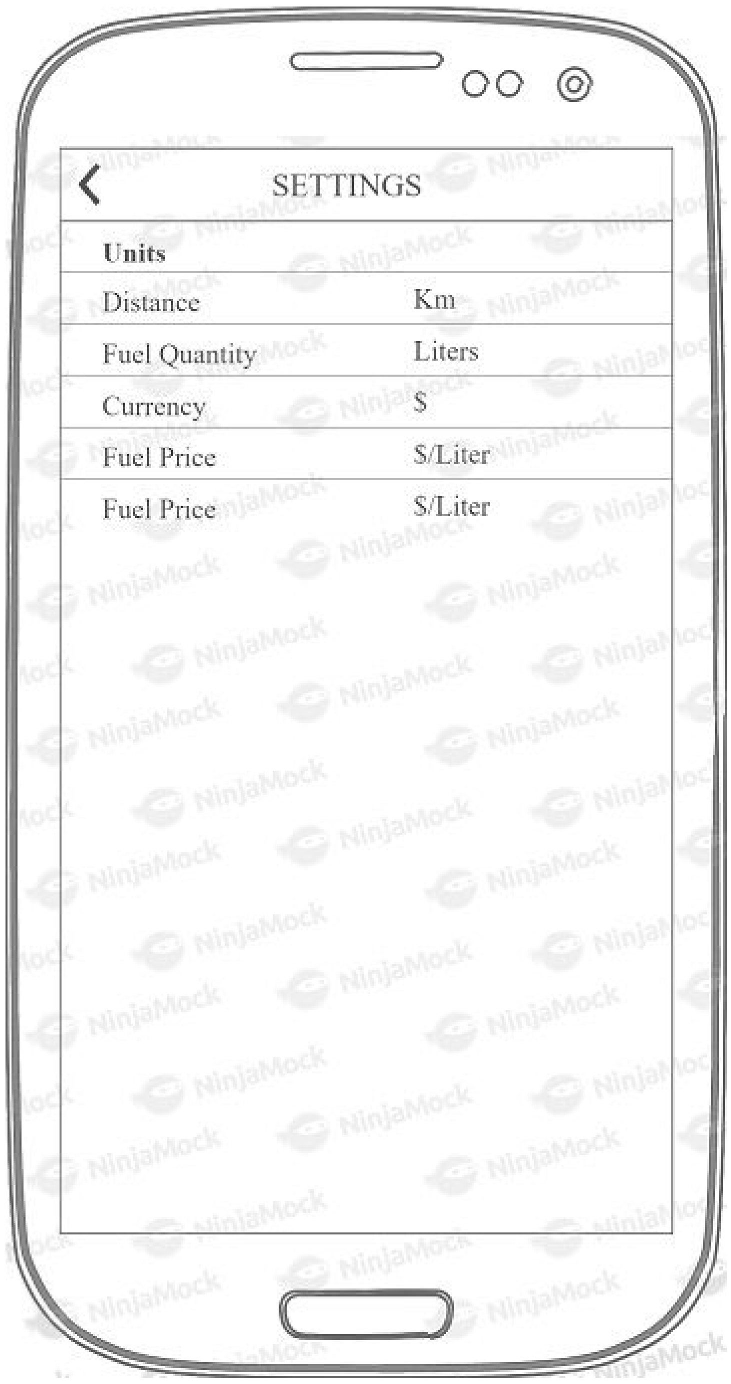
Screen 4



Date	Text Field
Car	Text Field
Trip Dist	Text Field
Quantity	Text Field
Price	Text Field
Total Price	Text Field
Fuel Type	Text Field
Note	Text Field

This is the refueling screen, it allows the refueling entry.

Screen 5

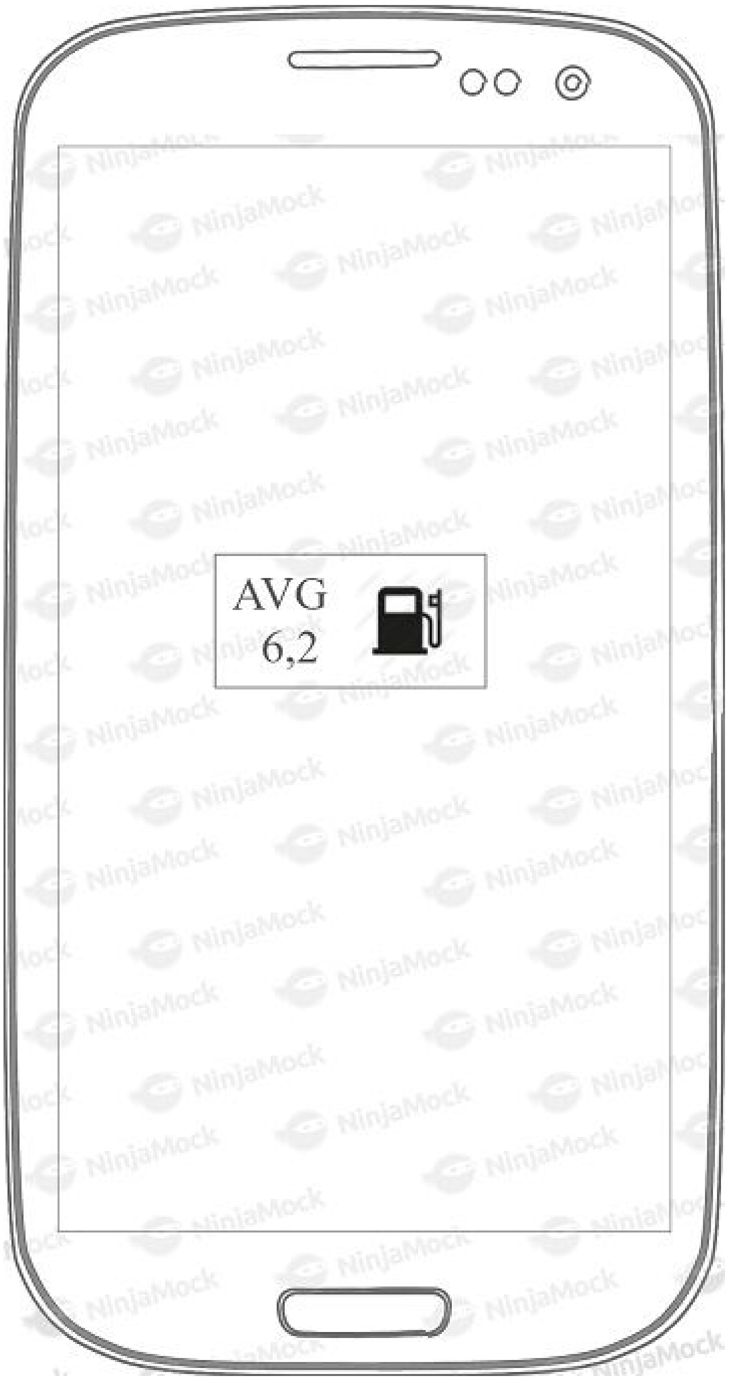


SETTINGS

Units	
Distance	Km
Fuel Quantity	Liters
Currency	$
Fuel Price	$/Liter
Fuel Price	$/Liter

This is the settings screen

Widget



This is the widget screen shows the average fuel consumption and and allows a direct entry refueling.

Key Considerations

Application will be written in Android Studio 3.1.3 solely in the Java Programming Language, gradle version 4.4

How will your app handle data persistence?

Applications will use Room Persistence and Firebase Realtime Database.
Insert() method Room library may block the main UI thread. Async Task will be used as a wrapper for this method.

How you will support accessibility?

Application will have a content description parameter for each view.

How resources will be stored in the project ?

All resources will be stored in the resource directory, for example: colors.xml, string.xml, theme.xml

Describe any edge or corner cases in the UX.

Change screen orientation: application save and read state and configuration correctly(e.g. save instance state or sqlite database).
Unstable network connection: application correctly supports the lack of an internet connection(in this case application read data only from sqlite database)

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

- Room Persistence Library to handle data persistence: 'android.arch.persistence.room:runtime:1.1.1'
- Android support library to provide newer features on earlier versions of Android: 'com.android.support:appcompat-v7:27.1.1'
- Support Library to create Card View: 'com.android.support:cardview-v7:27.1.1'
- Support Library to create Recycler View: 'com.android.support:recyclerview-v7:27.1.1'
- Android support design library to provide material design on earlier version of Android: 'com.android.support:design:27.1.1
- MPAndroidChart to easy use chart: 'com.github.PhilJay:MPAndroidChart:v3.0.3'
- UI Tests Espresso: 'com.android.support.test.espresso:espresso-core:3.0.2'

- Firebase RealTime Database: 'com.google.firebase:firebase-database:16.0.1'

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

- To use Firebase RealTime Database: 'com.google.gms:google-services:4.0.1',
- To display Ads: ’com.google.android.gms:play-services-ads:15.0.1’

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

Create new project in Android Studio and configure gradle files:
- Select minimum SDK
- Update Gradle and add the required libraries and dependencies

Task 2: Implement UI for Each Activity and Fragment

Build User Interface:
- Build UI for MainActivity
    - Create Card View
- Build UI for RefuelingListActivity
    - Create recycler view
- Build UI for ChartActivity
    - Add MPAndroidChart library to gradle dependency
    - Add Charts to view
- Build UI for RefuelActivity
- Add a content description parameter for each view (support accessibility)

Task 3: Data persistence

Create SQLite database with abstraction layer Room Persistence Library:
- Create database schema
- Add required dependencies
- Create the entity classes
- Create the DAO classes

- Create the LiveData class
- Implement the Room database
- Implement the Repository
- Create the ViewModel
- Create AsyncTask to insert data
- Connect with UI

Task 4: Cloud data persistence

Implement Firebase Realtime Database:
- Add required dependencies
- Create Firebase project in Firebase console
- Include the google-services plugin
- Retrieve an instance of your database

Task 5: Ads

Implement AdMob ads:
- Implement google play services
- Initialize MobileAds with AdMob App Id
- Add Ads to UI

Task 6: UI Tests

Create UI Test with Espresso framework:
- Add required dependencies with test annotation
- Writing several espresso tests for different views

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**"