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: HarunJr

CashTrack

Description

CashTrack is a record taking app that takes cash collected during a bus trip (from point A to B) and stores it in a remote server as well as calculates the daily total income and expenses per vehicle.
It is necessary because the current mode of taking records is paper based and thus slows down daily closing of business for many Public Service Vehicles (PSVs) in Kenya. CashTrack gives real time updates of how cash is flowing in making it available for the bus owner/manager.

Intended User

This app is for on ground operators known as '*off-loaders'* who collect cash from passengers in psvs as the market is cash based. As vehicles transport passengers from the Nairobi Central Business District (CBD) to other districts within Nairobi at various bus stops, the off-loaders' main job is to collect cash and keep the owner of the vehicle updated on cash flow per trip. This app therefore helps off-loaders record all cash transactions for a given vehicle, and for the vehicle owner to have access to these real time updates time per trip. This eventually helps the vehicle owner keep a detailed cash flow record and assist in general management to the vehicle.

Features

- Create User
- Add Vehicles
- Add Collection/Expenses
- Display total Collection and Expense per vehicle
- Gets balance per vehicle
- PIN authentication
- Periodic Notification Reminders

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

This is the first page met by the user.
It takes in the mobile phone number of the user.

Screen 2

After a user enters their phone number, they are prompted to create a four digit PIN.

Screen 3

Next, the user confirms his PIN

Screen 4

This is the default ‘login page’, takes in the pin for authentication.

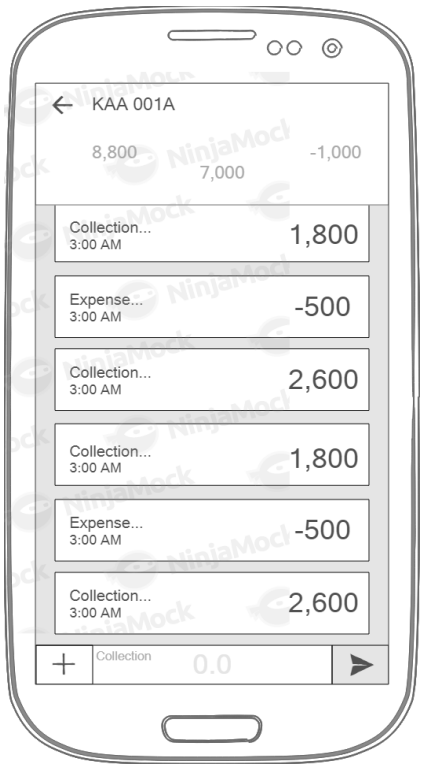
Screen 5

The vehicles Page is the main page consisting a list of vehicles the offloader is supposed to collect cash from.

Screen 6

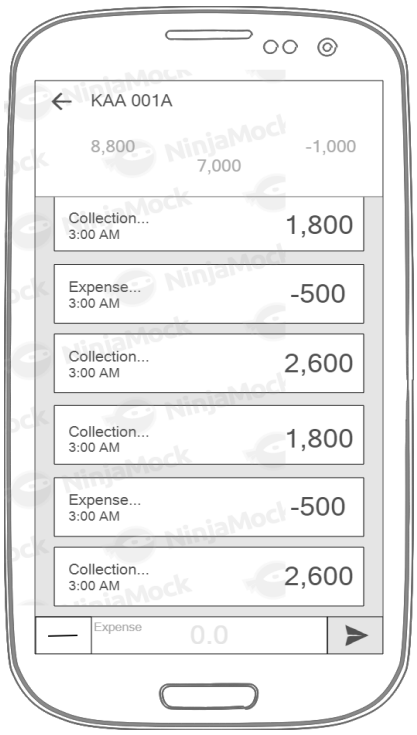
Here, the user adds a vehicle details.
It’s accessed through the ‘ADD VEHICLE’ menu item.

Screen 7



This is the details page containing a list of all transactions for the specific vehicle,
total collection for the day, total expense for the day and the difference at the top.
There's a textview at the bottom of the page for entering the transaction for the time and a plus and send button.
The 'PLUS' button when clicked changes to '-' expense.

Screen 8



On clicking the PLUS button on screen 7, it changes as above to show expense.

Screen 9

Clicking on the EditText box reveals a numeric Keypad from below.

Screen 10

A dialogue box opens when the SEND button is clicked on screen 9.
Clicking OK opens the next screen.
This is specifically for expenses, after entering collection, the app skips this screen to the next.

Screen 11

Snackbar displayed giving the user time to change his mind.
Clicking on UNDO, takes us back to the input page.
Sliding the snackbar left/ letting it time out closes the keypad.

Key Considerations

How will your app handle data persistence?

The app stores data locally using an SQLite database containing two tables;

- Vehicles
- Transactions

Data access for these table is managed using a Content Provider.
Shared Preferences may be used to store user data.

The app also stores the data in a remote database and managed using Retrofit.

The app uses Loaders to load data from SQLite to the views.

Describe any edge or corner cases in the UX.

The loginActivity consists of three fragments (LoginFragment, CreatePINFragment, ConfirmPINFragment) .If there is no user data in sharedPreference, CreatePINFragment is called, followed by ConfirmPINFragment, else, LoginFragment.

User must enter PIN each to access the data as it should be ‘sensitive information’

Clicking on the send button for the collection/expenses responds differently. An expense prompts the user to give a description whereas clicking ‘send’ when collection is selected skips this step and directly displays the snackbar.

When the user hits the ‘back’ button from either the Add vehicle fragment or the DetailsFragment, the user is taken back to the vehicle fragment.

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

Appcompat - ‘com.android.support:support’, ‘com.android.support:appcompat to support backward compatibility.

Design library - ‘com.android.support:design’ to enable use of coordinatorlayout and collapsingToolbarLayout.

Recyclerview - ‘com.android.support:recyclerview’ to display vehicles and transactions.

Retrofit - ‘com.squareup.retrofit2:retrofit:2.1.0’ , ‘com.squareup.retrofit2:converter-gson:2.1.0’to act as a HTTP client.

OkHttp - ‘com.squareup.okhttp3:logging-interceptor:3.4.1’, for sending and retrieving HTTP-based network requests.

Espresso for UI tests

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

Analytics : To see how users are using the app and how many users are using it as well as firebase crash reporting to be able to access the crashes that are happening on my app.
Authentication : Use of FirebaseUI Auth for phone number authentication.

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

- Design the wireframes for all the pages of the app.
- Setup Android Studio and create a new project
- Configure the required dependencies in the build.gradle file and ensure that there are no errors.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
    - LoginFragment
    - Create PIN Fragment
    - Confirm PIN Fragment
    - All layouts involved

- Build UI for MainActivity
    - Toolbar
    - Menu Items
    - Recyclerview for Vehicle Fragment

- Build UI for AddVehicleActivity
    - EditText Fields
    - Create Add Button, button

- Build UI for DetailsActivity
    - Recyclerview for Details Fragment
    - Custom keyboard
    - EditText
    - '+' and '-' button above the keyboard
    - Send button

Task 3: Data Persistence

- Build Database contract and Database Helper class
- Build a content provider for saving and retrieving data
- Create a DAO class for users, vehicles and transactions
- Build a loader to update vehicles in vehicles fragment

- Build a loader to update transactions in details fragment

Task 4: Network Connection

Create an Intent Service class to access data from a remote database i.e http://transit.gemilab.com API via Retrofit.

Create authentication from remote server

- http://transit.gemilab.com/users to create a new user
    - Input name
    - Input phone number
    - Input role i.e (offloader)

Allow access to the app using data from the API persisted by Shared Preference
Create vehicles by sending a POST request to http://transit.gemilab.com/vehicles
Add transactions by sending a POST request to http://transit.gemilab.com/transactions
Get vehicles from http://transit.gemilab.com/vehicles
Get transactions from http://transit.gemilab.com/transactions

Create a check for network connectivity before calling the data retrieval/send functions.

Task 5: UI testing.

Test aspects of the UI using espresso.

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**"