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: jasonlevinsohn

L3 Egg Sales

Description

Displays the sales numbers for our egg business at Llamas on the Loose Farm. User can choose sales numbers with time ranges of weekly, monthly, or yearly.

The app also has the ability to add new sales receipts to the Quickbooks ledger. This can come in real handy when a sale has just been made, and can be added to the Quickbooks ledger with a simple form in the app without having to remember the sale and add it in Quickbooks later on when doing the books.

Intended User

Employees of Llamas on the Loose Farm who sell our eggs. Right now, it is difficult to remember to enter all the egg sales into Quickbooks manually sometimes days or weeks after a sale. Since we always have our phones on us, this app will make it much easier to log the sales into our accounting and not have to remember or worry about it when we finally get around to opening up the books.

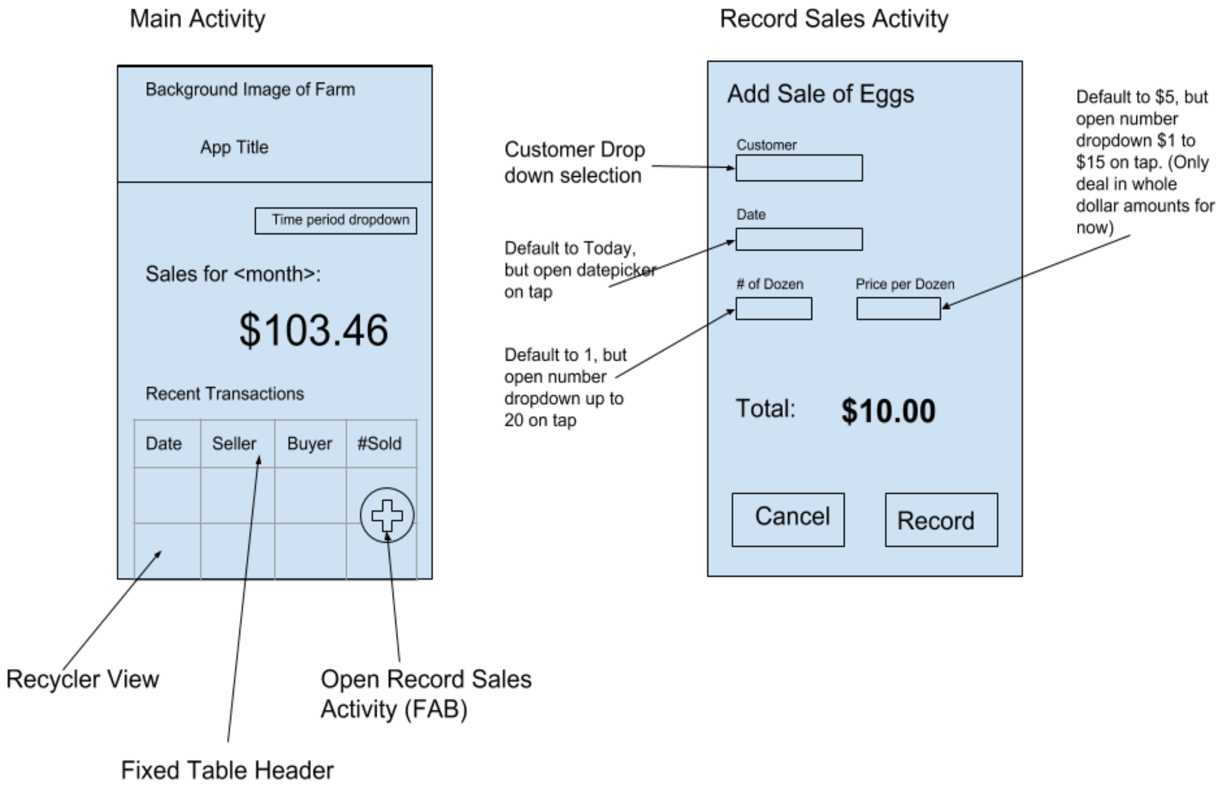
Features

- The app will use the Java language for development purposes.
- The app will utilize all stable release versions of libraries, Gradle, and Android Studio
  - Android Studio v. 3.1.2
  - Gradle v. 5.2.1
  - Android Support AppCompat v7:28:0:0
  - Android Support Constraint Layout v. 1.1.3
  - Android Support Design v 28.0.0
  - Facebook Stetho v. 1.5.1
  - JakeWharton Timber v. 4.7.1
- Displays current egg sales for current month by default from Quickbooks Account API
- Displays recent sales records
- User can choose a different time period from a dropdown list
- Shows last known sales data from local database upon opening the app, and updates data after a network call from the Quickbooks API has been received.
- Easy to use Sales Receipt form to add new sales to the business ledger

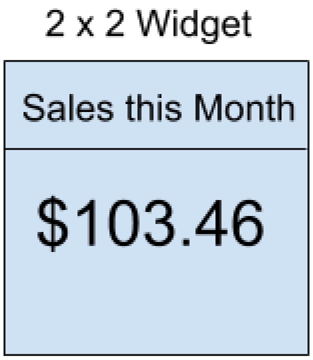
User Interface Mocks

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 and Screen 2


Main Activity
Background Image of Farm
App Title
Time period dropdown
Sales for <month>:
$103.46
Recent Transactions
Date
Seller
Buyer
#Sold
Recycler View
Fixed Table Header
Open Record Sales Activity (FAB)
Record Sales Activity
Add Sale of Eggs
Customer
Date
# of Dozen
Price per Dozen
Total:
$10.00
Cancel
Record
Customer Drop down selection
Default to Today, but open datepicker on tap
Default to 1, but open number dropdown up to 20 on tap
Default to $5, but open number dropdown $1 to $15 on tap. (Only deal in whole dollar amounts for now)

Widget



Key Considerations

How will your app handle data persistence?

The app will use the Firebase Realtime Database for data persistence. I already have an account setup with Firebase, it can handle data caching when the device is offline, and last but not least my company already uses Firebase in our production app so I will get some experience using it when I work on our production android app.

Describe any edge or corner cases in the UX.

On the Record Sales Activity, the user could also press the back button to cancel the recording of a sales transaction.

On the Main Activity, in the case where there are no sales for the time period selected. The interface will display “No Sales Recorded” text instead of the recycler view of data.

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

Stetho - for developing and testing network calls and database persistence
Timber - for easier debugging

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

Firebase Realtime Database - for data caching and persistence of UI data from Quickbooks api
Notifications - to let everyone else using the app a sale has been made

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 a new project with an API target of 23. I'm choosing this API level as it is the minimum SDK level for my company's production app (of which I will be working on soon) and the only users of this app will be employees of our farm which all have phones above 23

- Phone and Tablet Form Factor
- Generate a Basic Activity with FAB for Record Eggs Form
- Add libraries to build.gradle file.
- Keeps all strings used in strings.xml and enables RTL layout switching on all layouts.
- Include content descriptions on all text fields and images for accessibility purposes.

Task 2: Implement UI for Each Activity and Fragment

Build UI for Main Activity

- Use a constraint layout to build out look and feel of the Main Activity
- Use tools namespace to put placeholders in place of real data for now
- Use colors.adobe.com to generate a nice color pallette for the app
- Put the colors in colors.xml
- Implement FAB to navigate to the Record Sales Activity
- Add content descriptions for all views
- Create dropdown for Weekly, Monthly, and Yearly data

Build UI for Record Sales Activity

- Use constraint layout again to build out look and feel of Record Sales Activity
- Use tools namespace to put placeholders in place of real data where needed
- Implement cancel button to go back to MainActivity
- Add content descriptions for all views
- Add dropdown for current customers in Quickbooks.
- Add datepicker to date field that defaults to the current day
- Add a number picker to the price per dozen, but default to $5
- Add a number picker to # of dozen eggs but default to 1

Task 3: Implement Quickbooks API

- Create an api xml file to put the quickbooks api and make sure to ignore it from the GitHub repo
- Setup Network Helper Class to get data from Intuit Quickbooks api
- Setup and configure Google Realtime Database service
- Setup JSON parsing class to parse the data
- Make sure to handle error cases for incoming data
- Pull some data from api, parse it, and save it to the Google Realtime database

- Setup Caching so the stale data will show up upon the app opening and starts getting fresh data.

Task 4: Show Quickbooks data on MainActivity

- Show egg sales data on MainActivity
- Build ListView or Recycler view to show 30 latest transactions
- Create view for when there is no data available for selected time period

Task 5: Implement Functionality on Record Sales Activity

- Populate Customer Dropdown from Quickbooks api
- Populate date field with current date
- Implement date picker for date field
- Implement NumberPicker for # of dozen field
- Implement NumberPicker for price per dozen field
- Implement calculation for Total text field
- Handle error scenarios for any empty fields
- Show a toast for any fields not filled in.
- Handle sending POST data to Quickbooks api when record button is pressed

Task 5: Create a Widget

- Create a widget which shows YTD sales figures
- Use an IntentService to update the widget whenever a new sale is made.

Task 6: Setup Google Play Services

- Setup the app to show a push notification each time a sale is made
- Setup the app with performance monitoring to get insight on how well it is performing.

Task 7: Setup Gradle Tasks

- Set up a installRelease build which has no logging for a production release
- Set up a signing configuration, keystore, and passwords

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**”