

How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Navigation drawer](#)

[Items to buy list](#)

[Item details](#)

[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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement Signin/Signup activity](#)

[Task 3: Data, data, data](#)

[Task 4: Implement UI for each activity](#)

[Task 5: Implement widget](#)

[Task 6: Notification](#)

[Task 7: Prepare for Play Market](#)

GitHub Username: [guziks](#)

FamilyCart

Description

Shopping list with notifications shared between family members.

Intended User

Families where some members go shopping while other members know what is required to buy.

Features

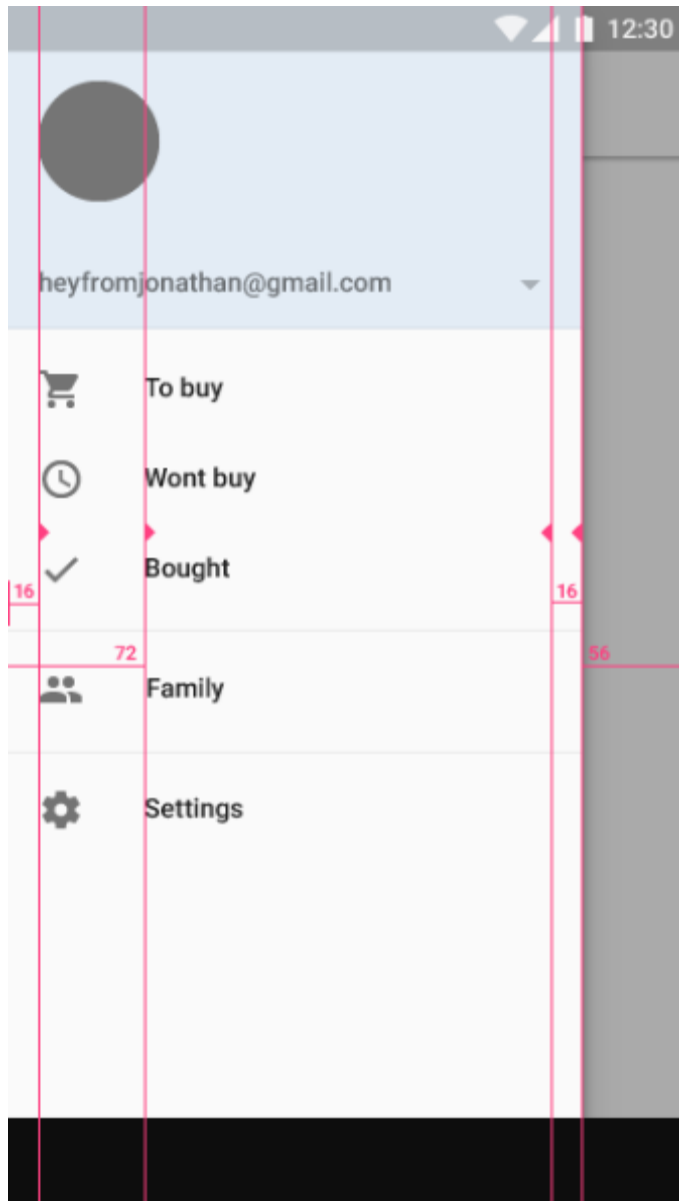
- single shared list of items "To buy"
 - swipe right to mark as "Bought"
 - swipe left to mark as "Wont buy": these items wont be recommended (recommendations - optional)
 - floating action button to add new item
- shared list of "Bought" items
- shared list of "Wont buy" items
 - swipe right to return to "To buy" list
 - swipe left to delete
- long tap (maybe some special area) and drag to rearrange
- long tap to enable contextual action mode and allow to manipulate multiple items (mark or delete)
- sort by:
 - name
 - date/time (last change)
 - auto (typical buying order) - optional
- details screen allows to:
 - enter name, quantity, extended comment
 - take picture (optional)
- uses Google Identity to identify family members
- add members one by one or choose existing G+ circle (e.g. "Family")
- uses Google Geofences to show notification if user came to typical shopping place but did not opened shopping list after several minutes (no notification if there are no items to buy)
- widget to display number of items left to buy

I'd like to implement "optional" features, but may not have time to.

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 Photoshop or Balsamiq.

Navigation drawer



Switch between lists is through navigation drawer

Item details

The screenshot shows a mobile application interface for editing item details. The screen is titled 'Item details' with a close button (X) on the left and a toolbar on the right containing a clock icon, a checkmark, and a menu icon. The form contains three input fields: 'Title' with the value 'Milk', 'Quantity' with the value '1 bottle', and 'Description' with the value '2.5% or 3% fat' followed by three lines of text: 'There is more space here', 'and here', and 'and here'. A large empty text area is below the description field. Red dimension lines indicate the following measurements: a vertical line of 16 on the left margin, a vertical line of 16 on the right margin, a horizontal line of 72 for the title field, and a horizontal line of 72 for the quantity field. A teal line with arrowheads at both ends spans the width of the description field.

Title	Milk
Quantity	1 bottle
Description	2.5% or 3% fat There is more space here and here and here

Three fields seen on previous screen are edited here. Also user can mark this item as “Bought” or “Wont buy” via toolbar actions.

Key Considerations

How will your app handle data persistence?

Create content provider to store all lists data offline.

Describe any corner cases in the UX.

Information entered in details activity should be saved (new item created or existing item edited) automatically upon “cross” tap.

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

For now it seems like no third party libraries required and it will be sufficient to use just Google Support libraries (RecyclerView features like ItemTouchHelper are essential to build list interactions).

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create general layout with navigation drawer and mock activities

Task 2: Implement Signin/Signup activity

- Refresh knowledge on Google Identity
- Create Signin/Signup activity

Task 3: Data, data, data

- Plan database structure
- Create content provider
- Create cloud endpoint
- Create sync adapter

Task 4: Implement UI for each activity

- family activity
- "To buy" list activity
- details/edit/new item activity
- "Wont buy" list activity
- "Bought" list activity
- check RTL support
- check accessibility

Task 5: Implement widget

- Create widget to show how many items left to buy

Task 6: Notification

- Implement geofence notification

Task 7: Prepare for Play Market

- Create Icon
- Sign application

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

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"