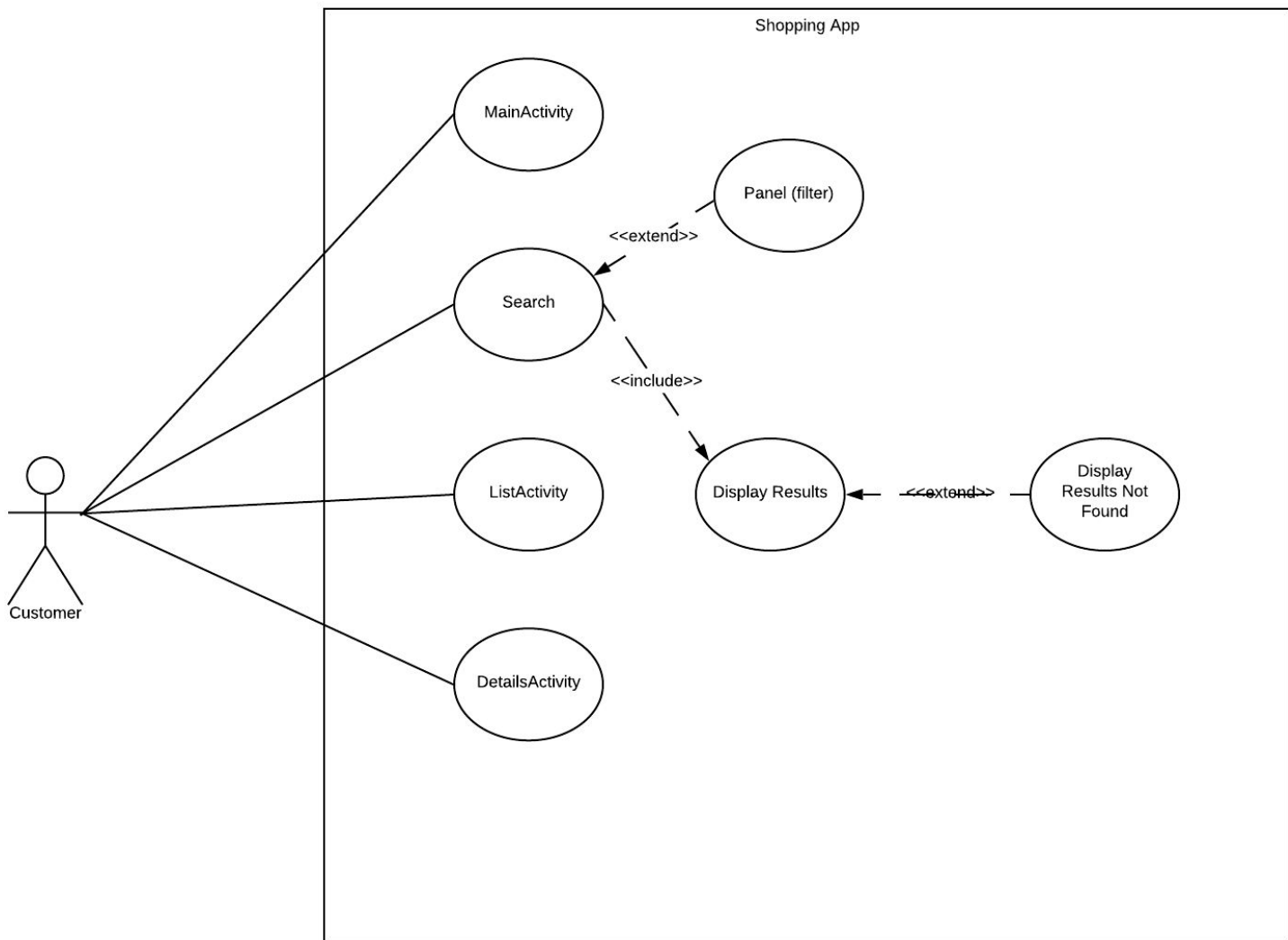


## Introduction

In COMPSYS 302 Project 2 (java) we will be creating an Android application to display Food listing items. There will be three sub-categories of Food items which are Sweets, Savoury and Drinks. This application will have simple functionalities including viewing a list of items available in each and all sub-categories of Foods. The user will be able to navigate through the application from the main page, to sub-categories and then to view a detailed individual item page. A search function will also be created for the user's ease of use. Searches would be able to be conducted using relevant filters.

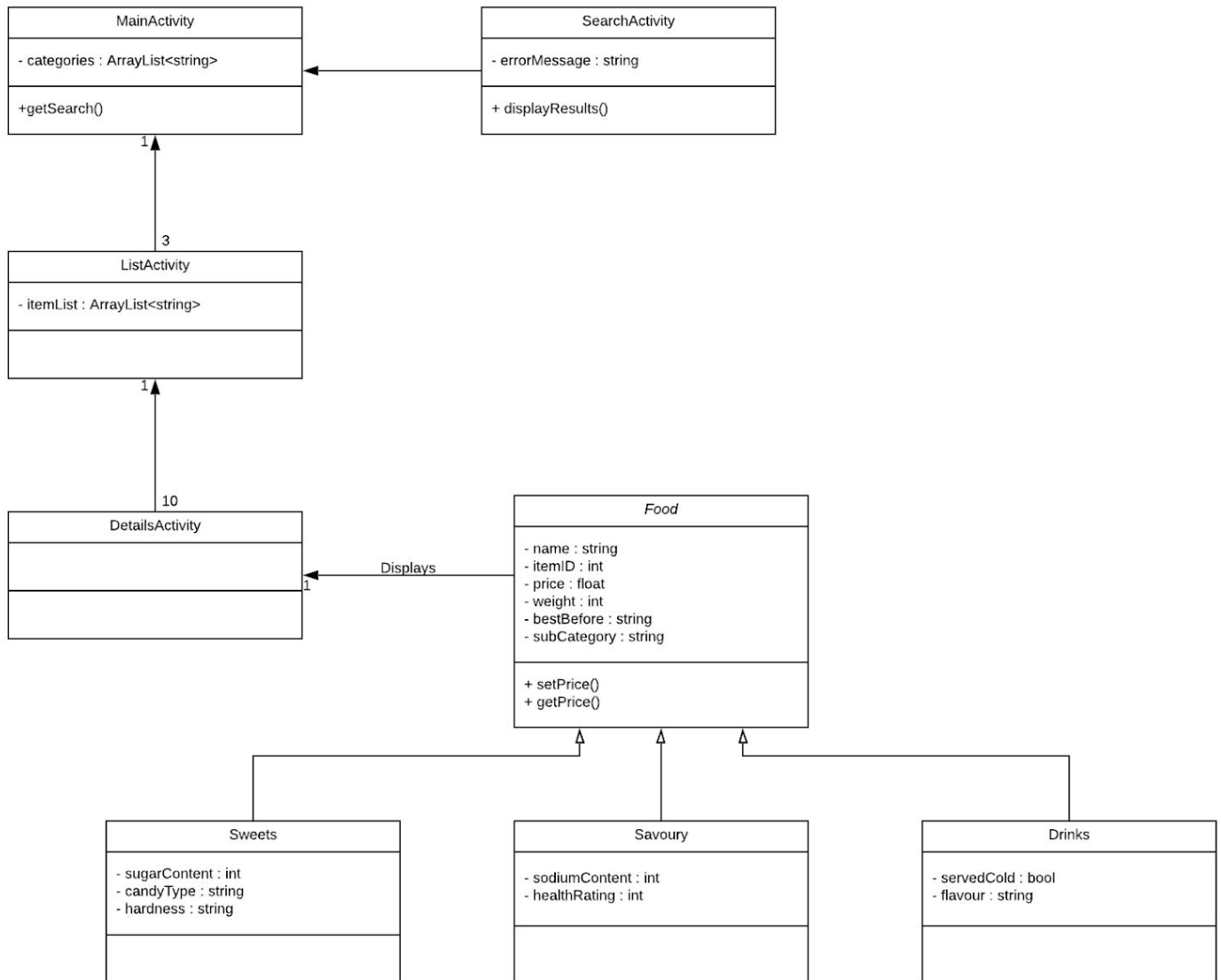
## System Modelling



There are four use cases in our modelling system which are MainActivity, Search, ListActivity and DetailsActivity which has an association relationship with our primary actor called 'Customer'. The user can go to each of these use cases directly by tapping on the search bar for the search function, tapping on the categories for ListActivity or by tapping on the Top Picks to proceed to the DetailsActivity directly.

The 'search' use case has an extended relationship from Panel (filter) which the user can sometimes access to filter relevant keywords (e.g bestselling, or most viewed, etc.) to reach their desired item. The 'search' use case also has an include relationship to the use case 'DisplayResults' which will always display something (i.e. options to reach the DetailsActivity use case) when the user types in a word. However, it will sometimes display 'Results not found' if the keywords do not match and therefore there is an extended relationship of 'Display results not found' to the use case 'Display results.'

## System Design

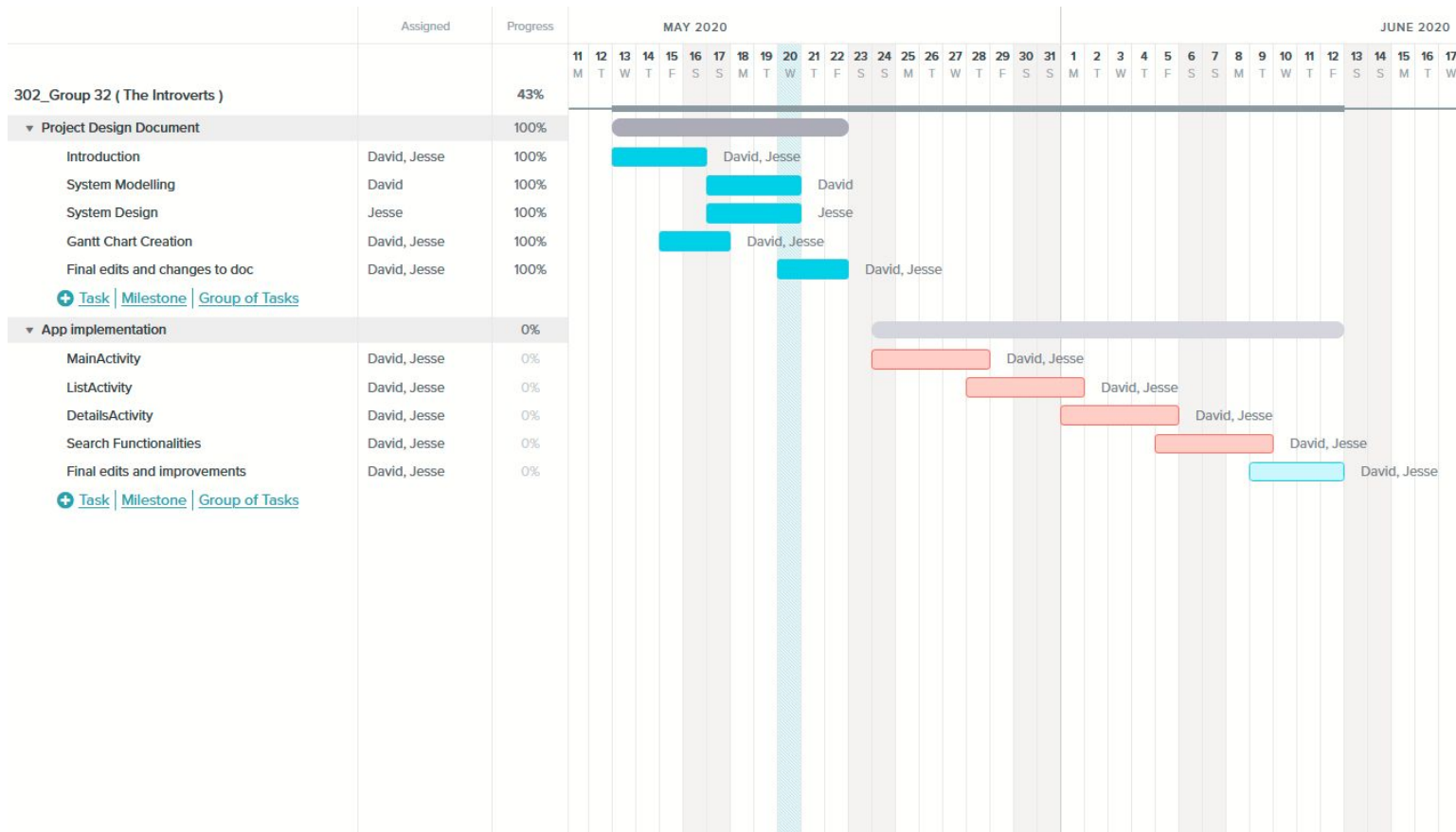


In our Android application, we will have MainActivity as the base class. Three categories of Food are shown and are all linked to different ListActivity (one for each category). A search can also be performed inside the MainActivity which is performed in the SearchActivity. Each of the three ListActivity displays 10 different items which are linked to a different DetailsActivity. One DetailsActivity displays the information of one 'Food' item as shown by the arrow in the class diagram.

We have a parent class of 'Food' which contains all the relevant private attributes of our child classes. The superclass needs to get and set the prices of the subclasses so we have the public methods 'setPrice()' and 'getPrice().'

The three child classes are 'Sweets', 'Savour' and 'Drinks' which inherit all the attributes of 'Food.' The individual child classes have their own private attributes as shown above.

## Schedule



This Gantt chart shows the responsibilities of each member of the team through the project. The chart and task deadlines were created by both team members in agreement. For this design document, David was in charge of the system modelling part (use case diagram) and Jesse was in charge of the system design (class diagram). The introduction and ideation for the project were done collaboratively.

In the second part of this project, the implementation will also be done collaboratively. This will be achieved through the use of GitHub where both of us will have access and publishing rights to the repository. Therefore it is reasonable to have both team members working on the same parts of the project at the same time. Although, we have decided that David will be mainly responsible for having the MainActivity and Search Functionalities polished while Jesse's main responsibility will be making ListActivity and DetailsActivity work flawlessly. Mini tasks within each section may also be divided during the course of project work.