

Development Plan

Andrew Johnstone: H00194083

Shayne Shaw: H00214626

Jack Rome: H00219766

Alex Birks: H00214362

Calum Thompson: H00187803

Chosen Collections

- HashSet – storing the menu when read in from a txt file.
- TreeSet – storing the collection of customers/orders.
- LinkedList – storing the items within an order.

Data Structure Justifications

The coffee shop's menu will be created from the text file read in at the beginning of the program. A *Menu* object will then be instantiated and its corresponding HashSet will be filled with the reported items. The *MenuItem* objects will be stored as the 'value' with the 'key' being the item's ID. In doing so, this will allow for easy referencing and searching of the available menu items. Another property of the HashSet implementation of hashTables, that will be beneficial to our application, is that they ensure that only unique entities are added. This helps prevent multiple entries of the same item being added to the menu if there is an error in the text file.

All *Order* objects that have been created during the running of the application will be stored in a TreeSet data structure within the *OrderList* object which. This is created when the text file is read at the start of the application. Each *Order* object will be stored as the 'value' and their corresponding UniqueID will be used as the 'key'. By using the tree implementation, a Set of *Order* objects, sorted using their UniqueIDs, can be returned. Allowing easy access to the *Order* object, either in order of appearance or by searching by a specific ID.

Each *Order* object created will have to store the items that a customer has ordered. A LinkedList will be used to do this. By storing the customer's order as they are taken, a LinkedList will allow for easier and more natural storing and recital. LinkedLists also allow for efficient insertion and deletion. Therefore, if a customer wants to remove an item previously added to their order, this can be done with minimal disruption. Another property of LinkedLists, that makes them suitable for this application, is that the nodes can be dynamically allocated. This means that a limit on the orders doesn't have to be set. They also allow for duplicate items to be added.

Sequence Diagram

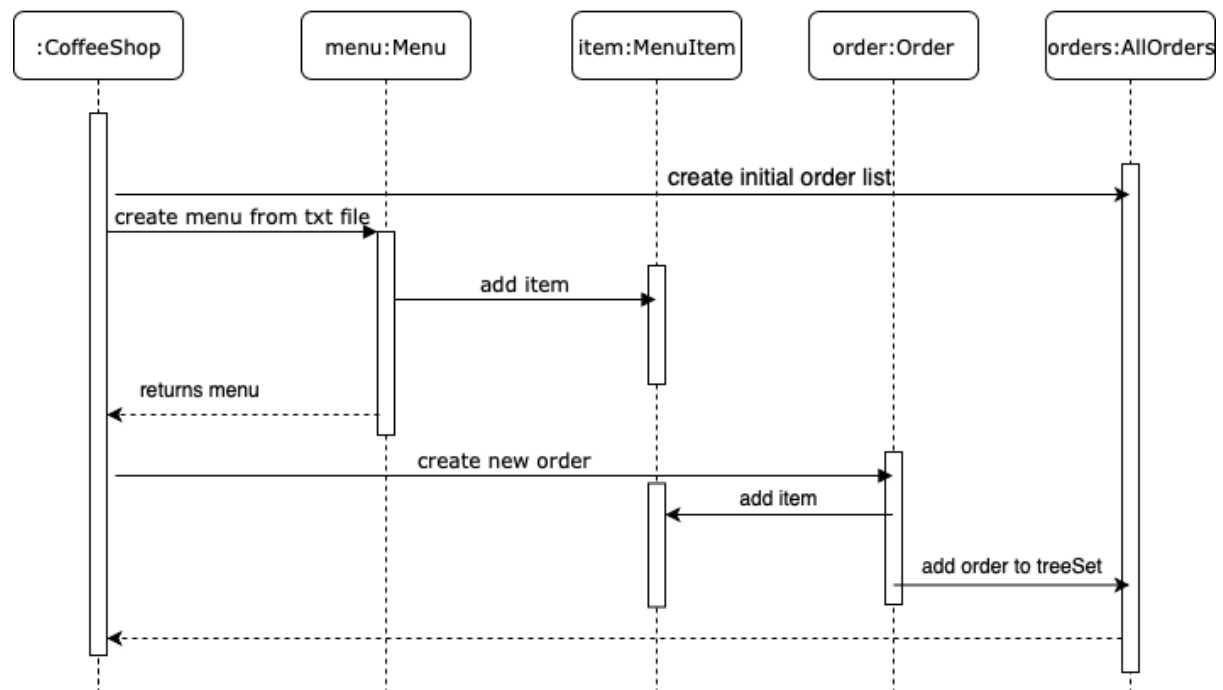


Figure 1: Sequence Diagram operation for importing menu and order files

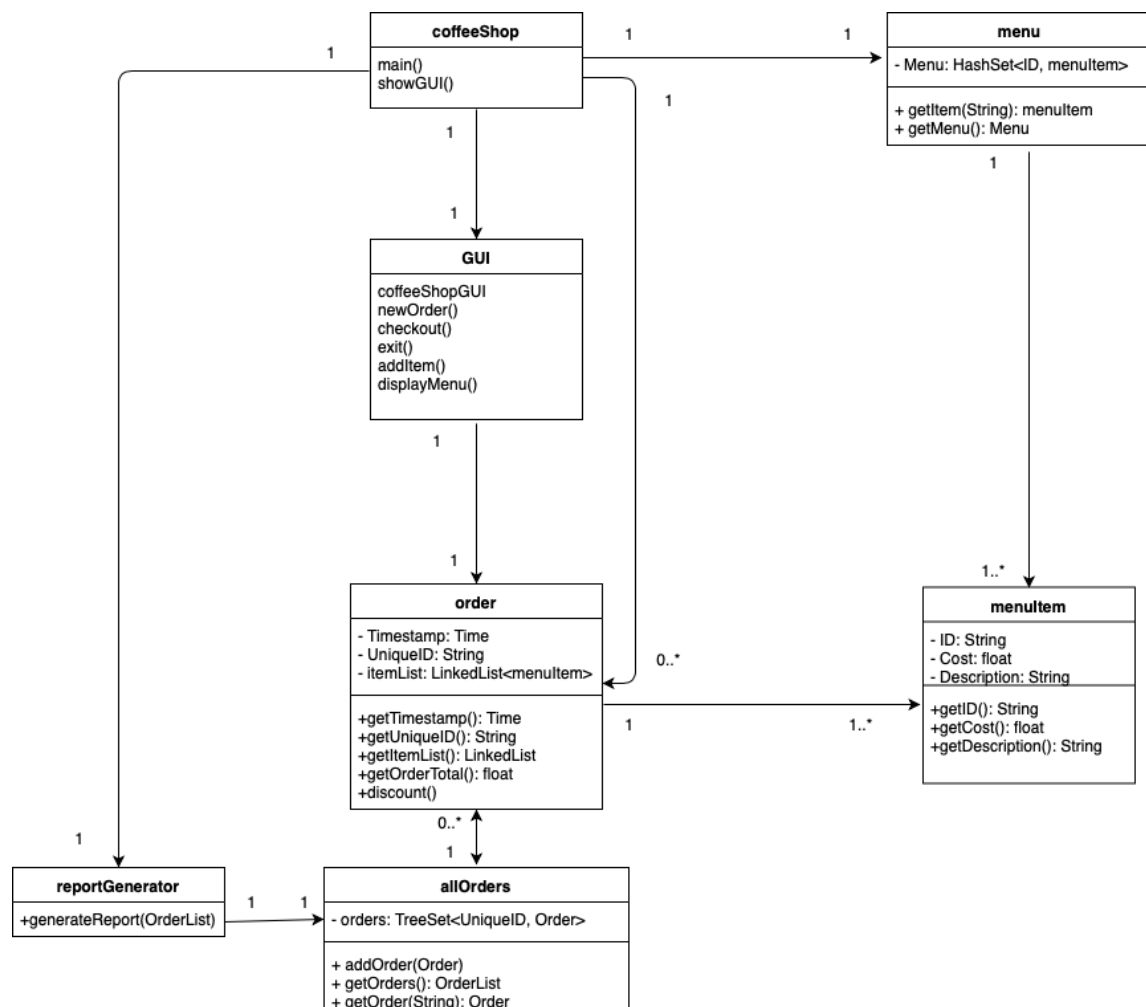


Figure 2: Final class diagram

Gantt Chart

