F21AS: Advanced Software Engineering

Group Report

Sabrina Chiesurin H00314757

Lorenzo James H00309663

Jowita Knap H00301565

Lucía Parga Basanta H00313273

1. Members of group and work. (add more).

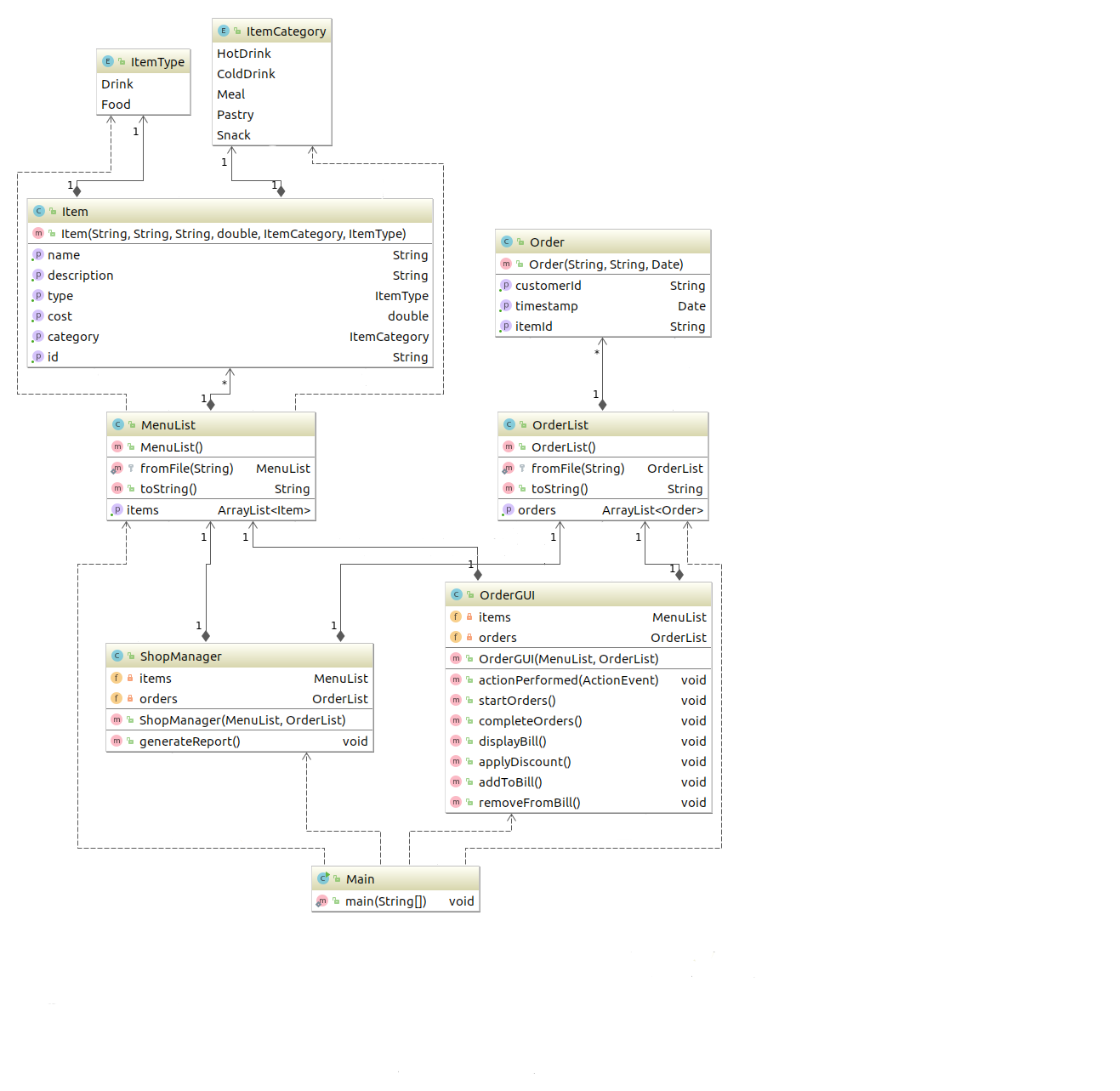
Jowita and Sabrina worked on the creation of the GUI and design of the application. Jowita started creating all the classes for and Lorenzo and Lucía continued working on the core of the application, developing the methods for the items, basket, discounts and the collections, Lorenzo worked on the import methods (text, read from text, creation of the menu and Lists, items), Lucía worked on the output part (implement methods, discounts). All of us decided about which type of Collections we were going to use and for what. About the Report,

1. Link to the repository: <https://github.com/lorenzo456/AdvancedSoftwareEngineeringCoursework>
2. This program meets the specification.
3. Suitable UML diagrams. (needs to be changed accordingly with the comments for previous development plan ).

Include Basket class, discount class

 CLASSES ENUMERATIONS  METHODS

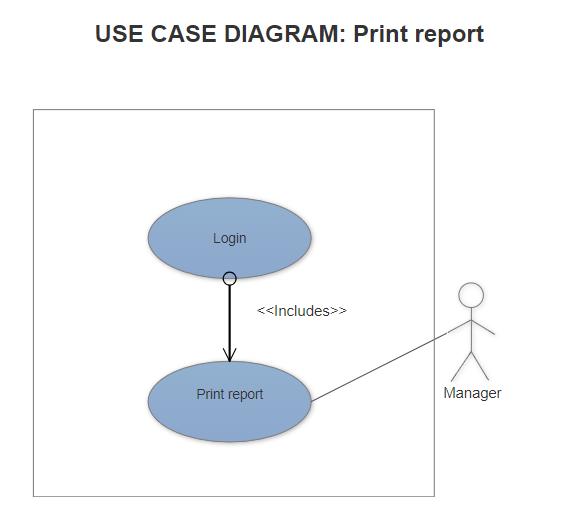
PROPERTIES  FILES  CONSTRUCTOR

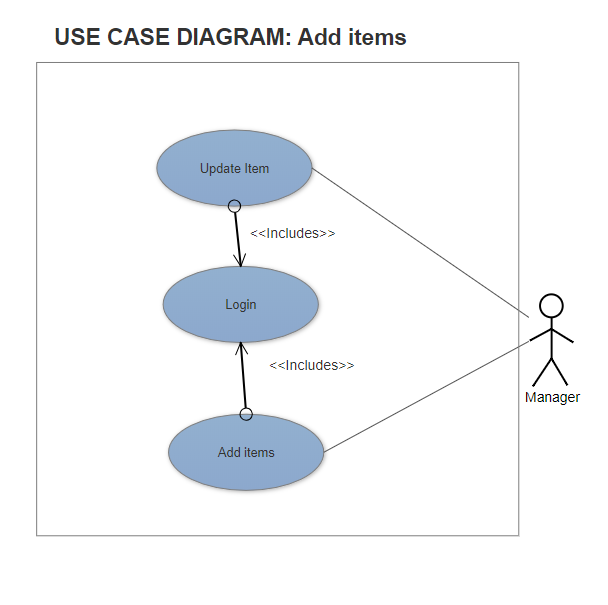


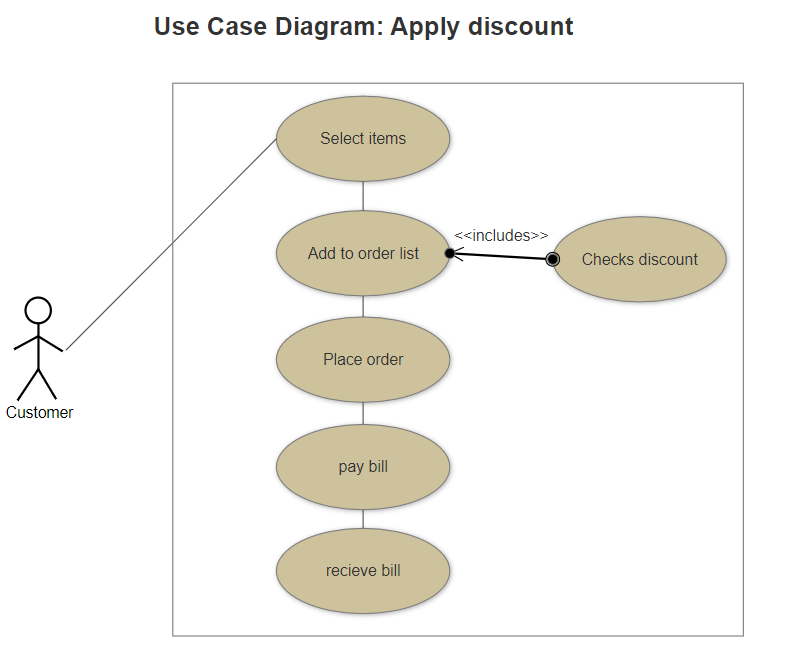
### Use Case Diagrams

Description of the possible interaction with the system that the different actuators will have in our Programme.

Include the normal worker, not only manager ☺ - he will check on discounts (Student, family).

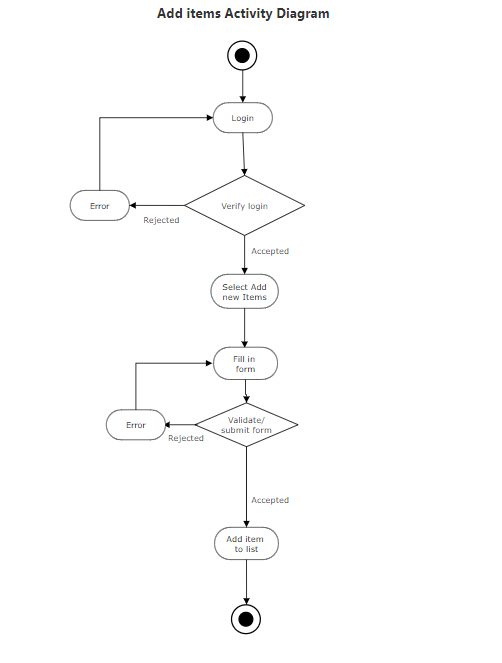
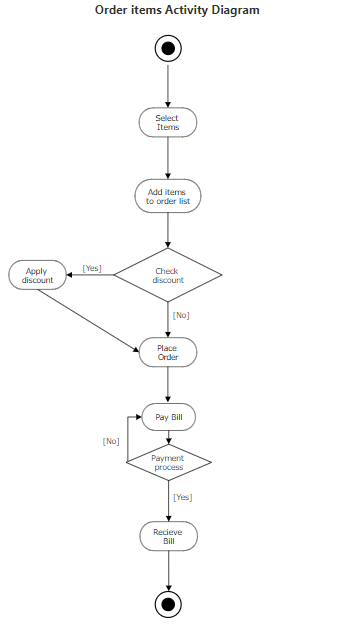






### Activity diagrams

Showing the workflow and how software and human activities interact.

1. Data Structures, which classes are used in and why we chose them.

**Basket** – LinkedList. The basket is going to store the elements the customer has selected to order. It will be stored as a linked list because we are going to be adding/removing elements frequently.

**Item list** –TreeSet. The item list is going to be storing the items the coffee shop has on sale. It will be stored in a tree set because the list will be searched through often, we want to display the items in order and we don´t want duplicated of the type items we have on sale.

**Discounts** – HashMap. The Discount list will store all the combinations of items which are on discount. It will be stored in a map, because each combination of elements need to have an unique key, we want to easily store combinations and don´t want duplicate discounts. Key will be the name of the discount and Value will be an array which contain the categories (as string) of the items included in that discount.

**DiscountList** – ArrayList. (Included in the Discount Class). This list will hold the values of the HashMap of the Discount class. We do not need extra specification with this List, so just and array is enough.

1. Decisions our group has made about the functionality of the program. (This is just information, not a technical report).

Why our programme works as it works ☺

1. Testing: What did you test using JUnit? Give details of which JUnit tests relate to which methods. Which types of exception did you use, in which method, and for what purpose?

To be done. Lucia and Lorenzo (try and catch). And Jowita (GUI).