|  |
| --- |
| USE CASE: Start Up  GOAL: Start up the till  ACTORS: Worker |
| **Pre-conditions**  The application is not on |
| **Main success scenario**   1. Server starts the application 2. The text file containing existing orders is read in and stored 3. The text file containing menu details is read in and stored 4. The GUI is generated with the menu details |
| **Exceptions**  2/3a. Error in reading the file   1. Fix the file path to the text file and continue from 1   2/3b. Identifiers are invalid   1. Manually revise the text file and continue from 1 |
| **Post-conditions**  The menu and existing orders are now stored in the application.  The GUI is displayed. |

|  |
| --- |
| **USE CASE:** Place Order  **Goals:** place an order to an item in the menu  **Actors:** Customer and worker |
| **Pre-conditions**  Application is switched and previous txt data has been loaded |
| **Main success scenario** |
| 1. Worker selects that they want to place a new order. 2. Customer indicates to worker what items they want to order. Worker browses and selects the items from display. 3. Bill is calculated and shown for the customer (sub use-case) 4. Customer pays either by cash or by card (requires card validation) |
| **Exceptions**  2a. Item is out of stock   1. Server is informed of the error, informs the customer and then continues from 2   2b. Customer changes their mind about the items they want to order   1. The worker deletes any unwanted items and continues from 2   4a. Credit card not validated   1. User is informed and continues from 4 |
| **Post-conditions**  The order has been placed and is stored in the system |
| **Sub use case**  Includes Calculate Bill |

|  |
| --- |
| USE CASE: Calculate Bill  GOAL: Calculate and display the bill  ACTORS: Server, Customer |
| **Pre-conditions**  The Application is running and the customer has ordered |
| **Main success scenario**   1. The total bill for all the items is calculated by summing the price for each item 2. If applicable a discount is applied e.g. student discount 3. The bill is checked for automatic discounts 4. If discounts are found they are taken off the total bill 5. The bill is displayed on the GUI showing the items purchases and their price, the total bill before the discount, and the total bill after the discount. |
| **Exceptions**  1a. Unexpected item on bill   1. Update the order in the use case Place Order (step 2) and continue to 1   4a. Applicable discount was not applied   1. Continue to 2 and apply the applicable discount |
| **Post-conditions**  The bill is displayed on the GUI |
| **Sub uses case**  Includes Place Order |

|  |
| --- |
| USE CASE: Close till  GOAL: To shut down till  ACTORS: Manager |
| **Pre-conditions**  The Application is running. |
| **Main success scenario**   1. The manager exits the application 2. The end of day report is generated and shown which will contain all of the items in the menu, the number of times each item was ordered and the total income before and after discounts. 3. The end of day report is saved on the system |
| **Exceptions**  3a. The report fails to save   1. Manager is told report was unable to be saved 2. Retry step 2 for a maximum of 3 times |
| **Post-conditions**  The end of day report is shown and saved as a text file |