RMIT
UNIVERSITY

COSC2081 – PROGRAMING 1

STORE MANAGEMENT

Nguyen Cong Thinh - *s3926387*

Le Ngoc Hieu - *s3927205*

Doan Duy Bach - *s3926953*
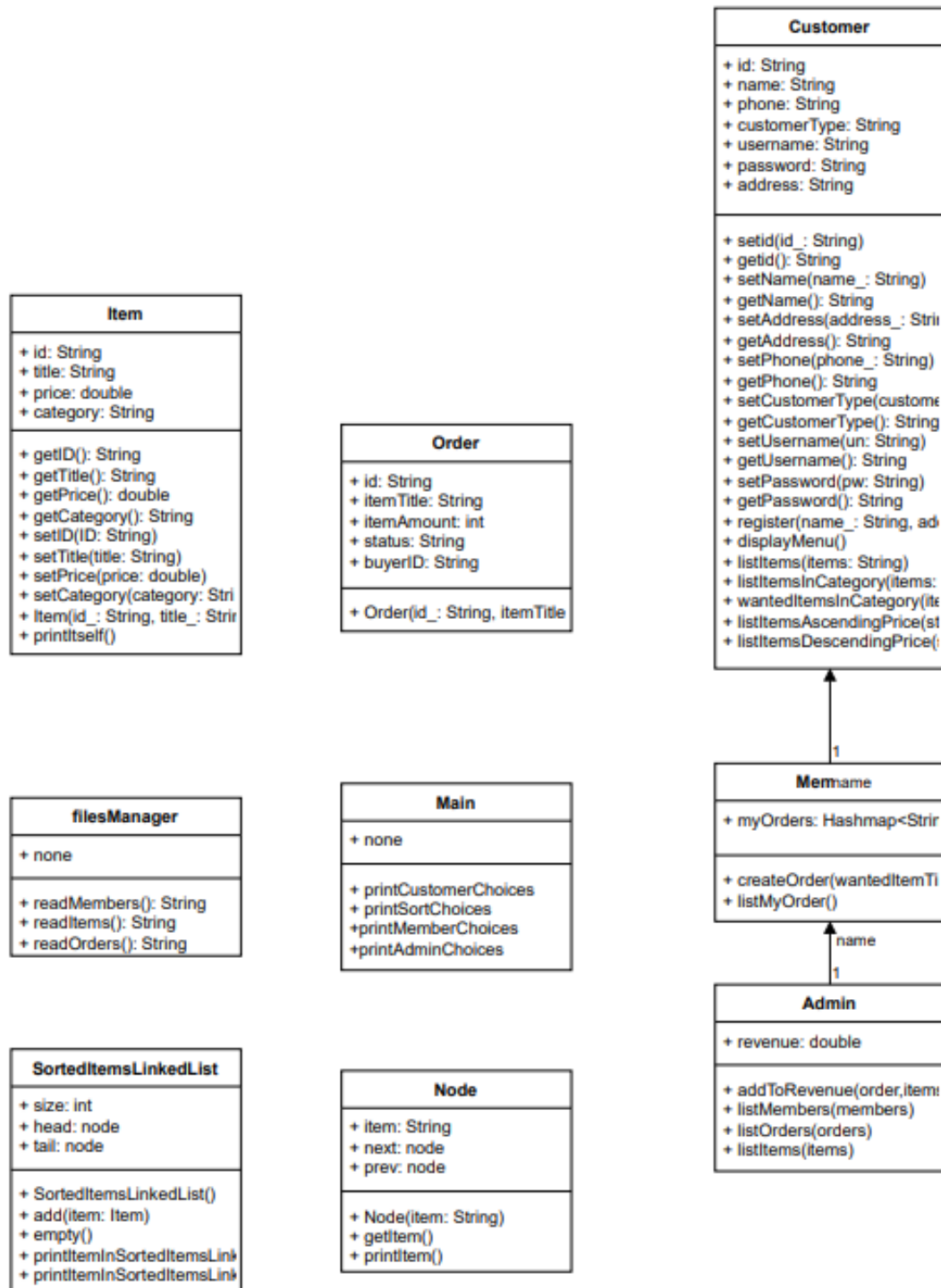
Don Tuan Duong - *s3927233*

1) Introduction:

Store management app is a project that is by far the most complicated project that any members of our group have faced, with Hieu and Bach as the main developers that build the core of the application. This project is a 2 weeks worth of work that we do our best to meet the requirement.

2) Project description:

This is a project that runs entirely by Java and it is an application which sole purpose is to run as store management. The app allows 3 kinds of users to log in and use it: Customer, Member, and Admin. Everyone that have logged into the app can view the information that they had used to log in. A customer could sort, filter, and search for an item or a list of items, a member could view information about a product and create orders to buy those items, and an admin is like a store owner that can add more products to the list or change the price of some items. Even if it was just a simple program with text-based UI, our group had made some personal touch to make it look more professional.

3) Implementation details:

Customer

+ id: String
+ name: String
+ phone: String
+ customerType: String
+ username: String
+ password: String
+ address: String

+ setid(id_: String)
+ getid(): String
+ setName(name_: String)
+ getName(): String
+ setAddress(address_: Stri
+ getAddress(): String
+ setPhone(phone_: String)
+ getPhone(): String
+ setCustomerType(custome
+ getCustomerType(): String
+ setUsername(un: String)
+ getUsername(): String
+ setPassword(pw: String)
+ getPassword(): String
+ register(name_: String, ad
+ displayMenu()
+ listItems(items: String)
+ listItemsInCategory(items:
+ wantedItemsInCategory(ite
+ listItemsAscendingPrice(st
+ listItemsDescendingPrice(

Item

+ id: String
+ title: String
+ price: double
+ category: String

+ getID(): String
+ getTitle(): String
+ getPrice(): double
+ getCategory(): String
+ setID(ID: String)
+ setTitle(title: String)
+ setPrice(price: double)
+ setCategory(category: Stri
+ Item(id_: String, title_: Strir
+ printItself()

Order

+ id: String
+ itemTitle: String
+ itemAmount: int
+ status: String
+ buyerID: String

+ Order(id_: String, itemTitle

Memname

+ myOrders: Hashmap<Strir

+ createOrder(wantedItemTi
+ listMyOrder()

filesManager

+ none

+ readMembers(): String
+ readItems(): String
+ readOrders(): String

Main

+ none

+ printCustomerChoices
+ printSortChoices
+printMemberChoices
+printAdminChoices

Admin

+ revenue: double

+ addToRevenue(order,item
+ listMembers(members)
+ listOrders(orders)
+ listItems(items)

SortedItemsLinkedList

+ size: int
+ head: node
+ tail: node

+ SortedItemsLinkedList()
+ add(item: Item)
+ empty()
+ printItemInSortedItemsLink
+ printItemInSortedItemsLink

Node

+ item: String
+ next: node
+ prev: node

+ Node(item: String)
+ getItem()
+ printItem()

As the diagram show, our three core class is: Admin, Member, and Customer. The Main class is the class that call the function from the other classes to run the home screen and the program. Some classes such as Item, SortedItemsLinkedList, Node, Order, and filesManagement are programs that read the given text files and turn the data into variables and also contain support functions.

4) Project planning report:

All team members	Role and tasks given	Individual contribution
1. Nguyen Cong Thinh	Make functions for the Customer	25%
2. Le Ngoc Hieu	Make the data-management function for the app	25%
3. Doan Duy Bach	Make functions for the Admin	25%
4. Don Tuan Duong	Make functions for the Member	25%

Right when the assignment was released, our group began to brainstorm the requirements. There sometime before the deadline, we decided to have an offline meeting to solve some of the difficult tasks that could hardly explain through online video chat and messenger. Our main method of communication is either through Discord meetings or Messenger video calls. Thanks to the group having a negotiation on the working and meeting time, we didn't meet any hiccups along the way.

Even though everything works smoothly, this project took much more time than we originally planned, we did feel nervous when the app keep having bugs when the deadline is near. At first, some of us are quite slow and not good enough to know what to do, so it took us a few days to learn and find information related to the assignment to make some basic functions of the app.

Timeline:
- 28/8: Planning and assigning tasks
- 29/8: Everyone starts to do the tasks that were given to them
- 13/9: Some basic functions have been completed, Functions for Admin and Member are in their final stage
- 15/9: The app is almost complete, checking for bugs and errors

- 16/9: The app is complete and going through the final examination
- 18/9: Complete the report and recording

5) Conclusion:

After thorough speculation, we did spot some drawbacks in this project, the code doesn't have a good coding practice, many variables are put in public. Without a diagram beforehand, it make confusing relationships among methods of interacting classes.

For future improvement, we planned to have a choice option for payment methods, shipping, and a working GUI instead of a text-based UI like what we have right now.