Term 1

Task 1: Project

PART 1 SUBMISSION

COLIN FERREIRA

11 COMPUTER SCIENCE: ATAR

Task 1: Community Store Software Development Project

Assessment type:

Project

Conditions:

Period allows for the completion of the task: 5 weeks.

Recommended time allocation
Part 1: 2 weeks for the completion of the investigation and design of the project.

Part 2: 3 weeks for the development and evaluation of the project

Task weighting:

20% of the school mark for this pair of units

Scenario

You have a friend that has recently started volunteering at a local Community Store. The Community Store accepts donations of clothing, including workwear, business attire, and casual clothes. It also accepts linen such as bedsheets and towels. To support people who are struggling in the community, the store has recently started accepting a range of household goods. These could include electrical items, kitchen utensils and crockery, and other homewares such as photo frames, storage items, etc.

Your friend has told you that the current system for processing what people are buying is difficult to manage and takes forever to process each purchase.

When processing a purchase, each item within each category has a fixed price (i.e. a pair of pants is $5, a button-up shirt is $3, and towels are all $3.50). The person processing a purchase needs to write down every item that is purchased, then consult the pricing guide, and work out the total. They then need to provide the purchaser with a receipt as many charities will reimburse some of the purchased items. The Community Store also requires a receipt for stocktake tracking and accounting purposes.

The Community Store would also like the ability to update prices and add new items, too, as they are going through a store transformation, and they may also introduce other sale categories such as food and pre-prepared meals.

You have recently been learning Python in your Computer Science class and think that you might be able to help with a simple, low-cost software solution. As part of the solution and once you've completed the main text-based program logic, you may like to explore integrating a GUI Python module such as Tkinter to allow for a simple graphical user interface. Another option is https://realpython.com/pysimplegui-python/.

Overview:

Your task is to use the software development framework to design and create a simple software application in Python that will help your friend in the Community Store.

To successfully develop the application, you will need to make use of:

- Variables and constants using appropriate naming conventions and data types.
- Data structures including one-dimensional arrays.
- A variety of control structures, including sequence, selection, and iteration
- A modular approach using functions and parameters, and argument passing.
- Programmatically writing data from Python to a text file.

Your final project needs to be complex enough to demonstrate your understanding of software development.

Part 1

Investigate

- Break down the steps required to produce your software project and develop a timeline for when each of these steps needs to be completed. You may like to use Trello to achieve this.
- Problem outline
  - Write a brief outline of the purpose of the application and its objectives.
- Problem description
  - Write a detailed description of the application that includes:
    - The structure of the application
    - Required sections to cover different operations.
    - How a cashier will interact with the application
    - How the system will handle each of the requirements from the scenario
- Create simple data structure design with a structure chart.
- Create a development schedule for the next 6 weeks of this project.

Design

- Using pseudocode, write an algorithm showing the core logic for the application to show how it will work and how a cashier will interact with the application. Note: this algorithm will not resemble your final, fully functioning code. It should simply demonstrate how the core logic for the application will work.

Project Outline and Objectives:

Project Title: Community Store Management System

Objective:

The objective of the Community Store Management System is to provide a **simple, efficient, and user-friendly** software solution to streamline the process of **managing purchases, maintaining inventory, and generating receipts** at a local Community Store. The system aims to replace the current manual process, which is time-consuming and prone to errors, with an automated solution that enhances operational efficiency and improves customer service.

Outline:

The Community Store Management System is designed to address the following key requirements and challenges faced by the store:

Automated Purchase Processing:

The app will automate the process of processing purchases by allowing cashiers to input purchased items, automatically calculating the total amount based on fixed prices for each item category and generating digital receipts for customers.

Inventory Management:

The app will maintain an up-to-date inventory of available items in the store, allowing staff to easily track stock levels, monitor product sales, and identify items that need to be restocked.

Price Updating and Item Addition:

The app will provide functionality for store administrators to update prices for existing items and add new items to the inventory. This flexibility will accommodate changes in pricing and support the introduction of new product categories as part of the store transformation.

Receipt Generation:

The app will generate digital receipts for customers, including details of purchased items, total amount, and any applicable discounts or reimbursements. These receipts will serve both customer needs and the store's requirements for stocktake tracking and accounting purposes.

Optional GUI Integration:

As an additional feature, the system will integrate a graphical user interface (GUI) using Python modules such as Tkinter or PySimpleGUI. This GUI will provide a visually appealing and intuitive interface for users, enhancing the overall user experience. This will be don’t for the sale module of the application, as it is the most utilised, as well as being an area where speed and efficiency is crucial.

By implementing the Community Store Management System, the store aims to improve operational efficiency, reduce processing time for purchases, enhance inventory management capabilities, and provide a better overall shopping experience for customers.

Problem Description

Project Title: Community Store Management System

Application Structure:

Main Menu:

The application will start with a main menu where cashiers can choose different options.
Options could include:

- Add New Item
- Update Prices
- Process Purchase
- Generate Receipt
- Exit

Add New Item:

- This section allows the cashier to add new items to the inventory.
- Cashiers will input details such as item name, category, and price.
- The system will update the inventory with the new item.

Update Prices:

- Here, the cashier can update prices for existing items in the inventory.
- Cashiers will input the item name or select from a list, and then input the new price.
- The system will update the price of the selected item in the inventory.

Process Purchase:

- In this section, the cashier will process a purchase for a customer.
- Cashiers will input each item purchased along with its quantity.
- The system will calculate the total amount based on the fixed prices and quantity of items purchased.
- It will generate a receipt for the customer, including the total amount due.

Generate Receipt:

- This section allows cashiers to print or generate a receipt for the customer.
- The receipt will include details of each item purchased, its price, and the total amount due.
- Additionally, it will include information required for reimbursement by charities, such as the store name, date, and signature.

Inventory Management:

- The application will maintain an inventory of all items in stock, including their categories and prices.
- Cashiers can view the current inventory to check stock availability.
- The system will allow for easy addition, removal, and updating of items in the inventory.

Stock Tracking and Accounting:

- The application will track the sale of items to maintain accurate stock levels.
- It will provide reports on sales, inventory levels, and revenue generated.
- This information will be useful for accounting purposes and for making decisions on restocking items.

Cashier Interaction:

- Cashiers will interact with the application through a text-based interface initially.
- They will navigate through the main menu using keyboard input and select appropriate options for each operation.
- Input prompts will guide them through adding new items, updating prices, processing purchases, and generating receipts.
- The system will provide feedback on successful operations and handle errors gracefully.
- Future development will lead to the use of GUI. Where the cashier will interact through either a mouse/keyboard and or touch.

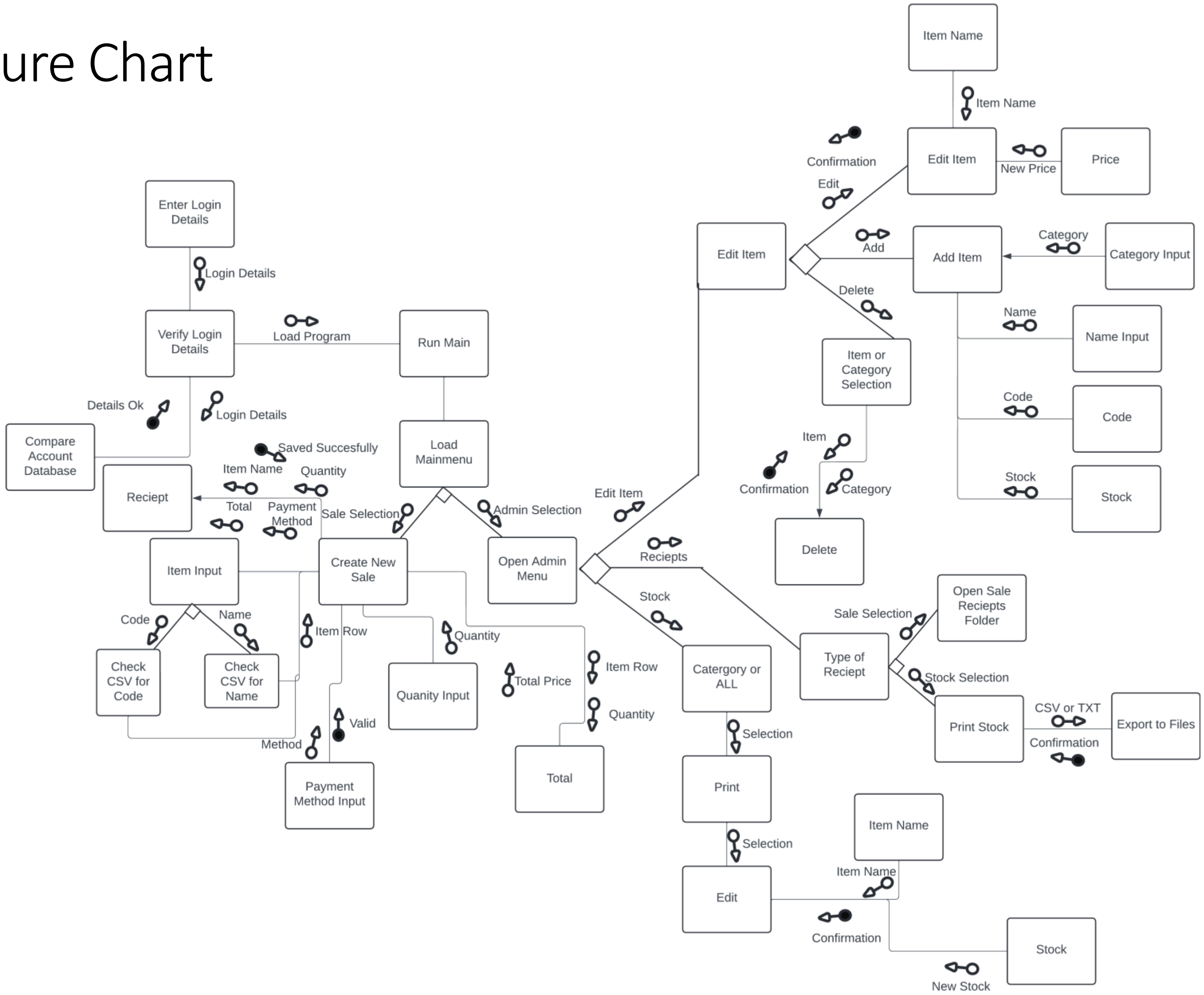
Handling Requirements:

- The application will have a modular structure, with each section handling specific operations independently.
- It will utilize file handling to store and retrieve data related to inventory, prices, and sales records.
- Regular backups of data will be performed to prevent loss of important information.
- The application will be designed to be scalable, allowing for easy addition of new features and sale categories as the store undergoes transformation.
- Integration of Tkinter or another GUI module will enhance user experience by providing a more intuitive interface for cashiers to interact with the application.

Pseudocode

Located in GitHub Repository

Structure Chart


Item Name
Item Name
Edit Item
New Price
Price
Confirmation
Edit
Edit Item
Add
Add Item
Category
Category Input
Delete
Name
Name Input
Item or Category Selection
Code
Code
Item
Confirmation
Category
Stock
Stock
Delete
Enter Login Details
Login Details
Verify Login Details
Load Program
Run Main
Details Ok
Login Details
Compare Account Database
Saved Succesfully
Load Mainmenu
Item Name
Quantity
Reciept
Total
Payment Method
Sale Selection
Admin Selection
Edit Item
Create New Sale
Open Admin Menu
Reciepts
Item Input
Stock
Sale Selection
Open Sale Reciepts Folder
Code
Name
Item Row
Quantity
Check CSV for Code
Check CSV for Name
Item Row
Catergory or ALL
Type of Reciept
Stock Selection
Total Price
Quanity Input
Quantity
Print Stock
CSV or TXT
Export to Files
Valid
Selection
Confirmation
Method
Payment Method Input
Total
Print
Selection
Item Name
Item Name
Edit
Confirmation
Stock
New Stock