

Wade Doolan

T1A3 - Terminal App

Presentation

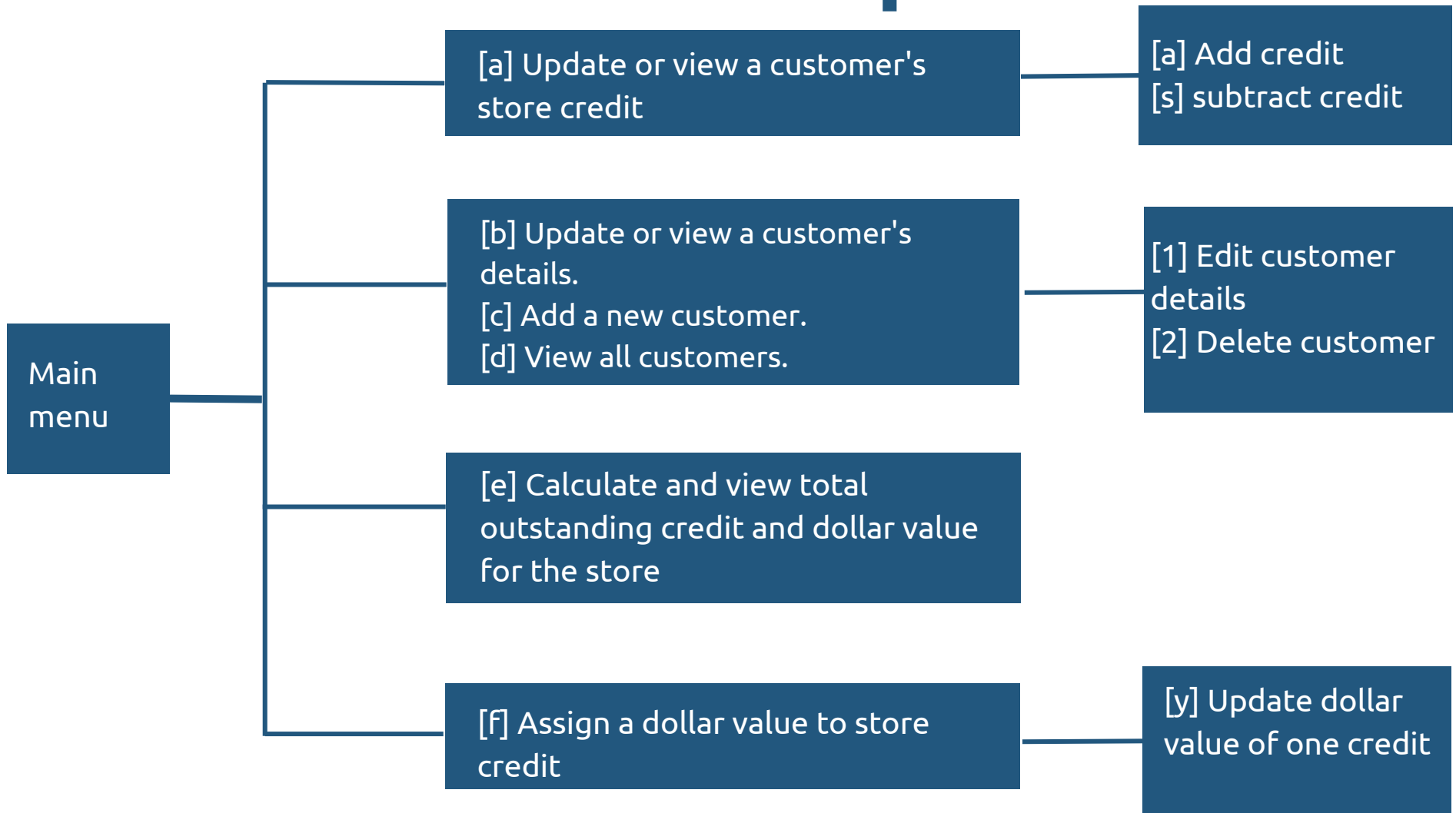
The Store Credit App

The Store Credit App allows retail staff in any retail setting to easily track customer store credit. The app has the following key features.

- A simple menu that allows retail workers to easily use the app, including access to a help document.
- Update or view a customer's store credit
- Update or view a customer's details. Add a new customer.
- Assign a dollar value to store credit
- Calculate and view total outstanding credit and dollar value for the store

Why this app? My brother runs a golf pro-shop and he wants an easier way to track store credit.

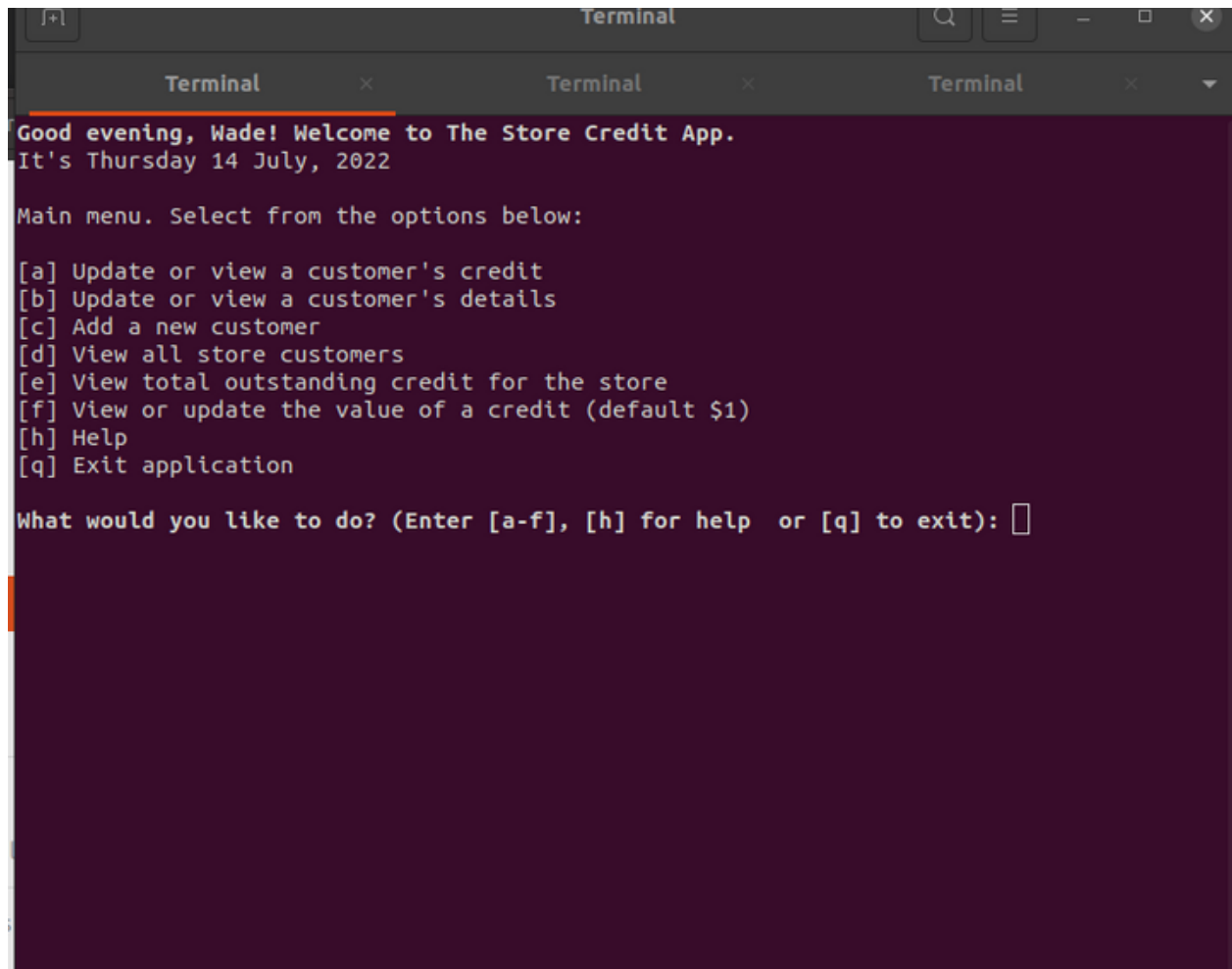
SiteMap



Application Walk-through

Main Menu

The application works around a main menu, which allows a user to carry out various tasks. Menu shown below:

A screenshot of a terminal window with a dark background and light-colored text. The window has a title bar with the word 'Terminal' and standard window controls. Below the title bar, there are three tabs, each labeled 'Terminal'. The main content area of the terminal displays the following text:

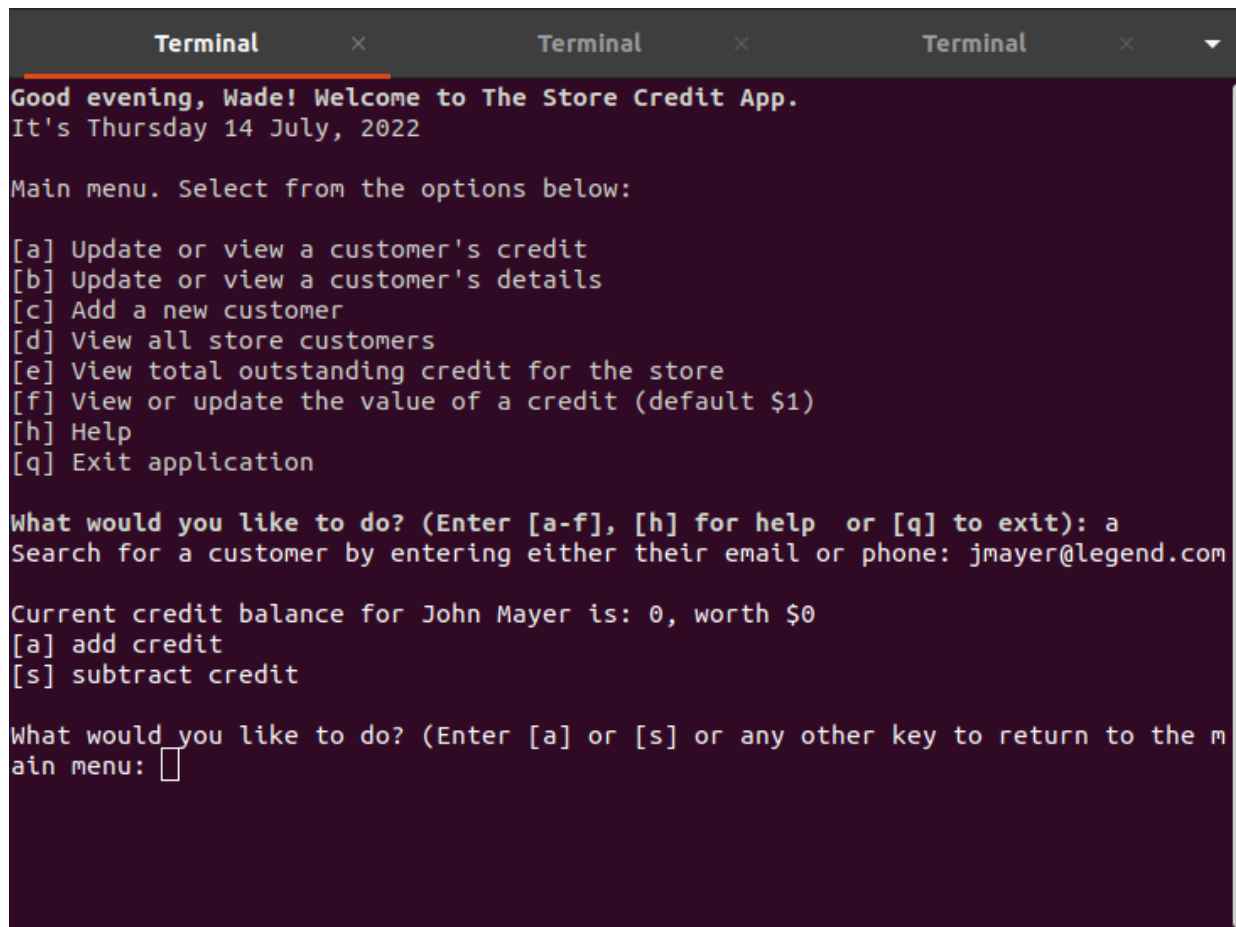
```
Good evening, Wade! Welcome to The Store Credit App.  
It's Thursday 14 July, 2022  
  
Main menu. Select from the options below:  
  
[a] Update or view a customer's credit  
[b] Update or view a customer's details  
[c] Add a new customer  
[d] View all store customers  
[e] View total outstanding credit for the store  
[f] View or update the value of a credit (default $1)  
[h] Help  
[q] Exit application  
  
What would you like to do? (Enter [a-f], [h] for help or [q] to exit):
```

The cursor is positioned at the end of the last line, ready for user input.

Application Walk-through

View/update customer credit

Selecting option [a] first prompts the user to search for a customer. When the customer is found the following menu is displayed:



```
Terminal x Terminal x Terminal x
Good evening, Wade! Welcome to The Store Credit App.
It's Thursday 14 July, 2022

Main menu. Select from the options below:

[a] Update or view a customer's credit
[b] Update or view a customer's details
[c] Add a new customer
[d] View all store customers
[e] View total outstanding credit for the store
[f] View or update the value of a credit (default $1)
[h] Help
[q] Exit application

What would you like to do? (Enter [a-f], [h] for help or [q] to exit): a
Search for a customer by entering either their email or phone: jmayer@legend.com

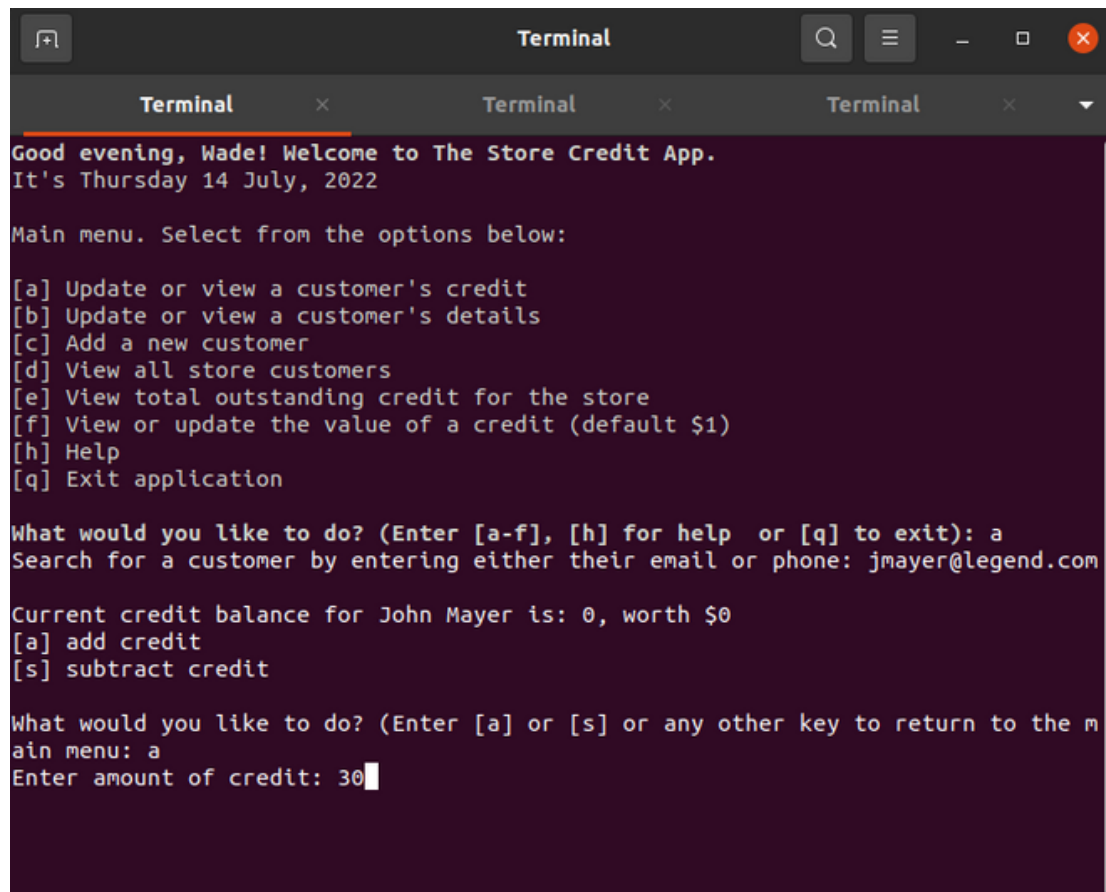
Current credit balance for John Mayer is: 0, worth $0
[a] add credit
[s] subtract credit

What would you like to do? (Enter [a] or [s] or any other key to return to the main menu: [a]
```

Application Walk-through

Adding customer credit

Selecting option [a] indicates the credit will be added against a customer's name. The user then enters the amount of credit to be added.



```
Terminal
Good evening, Wade! Welcome to The Store Credit App.
It's Thursday 14 July, 2022

Main menu. Select from the options below:

[a] Update or view a customer's credit
[b] Update or view a customer's details
[c] Add a new customer
[d] View all store customers
[e] View total outstanding credit for the store
[f] View or update the value of a credit (default $1)
[h] Help
[q] Exit application

What would you like to do? (Enter [a-f], [h] for help or [q] to exit): a
Search for a customer by entering either their email or phone: jmayer@legend.com

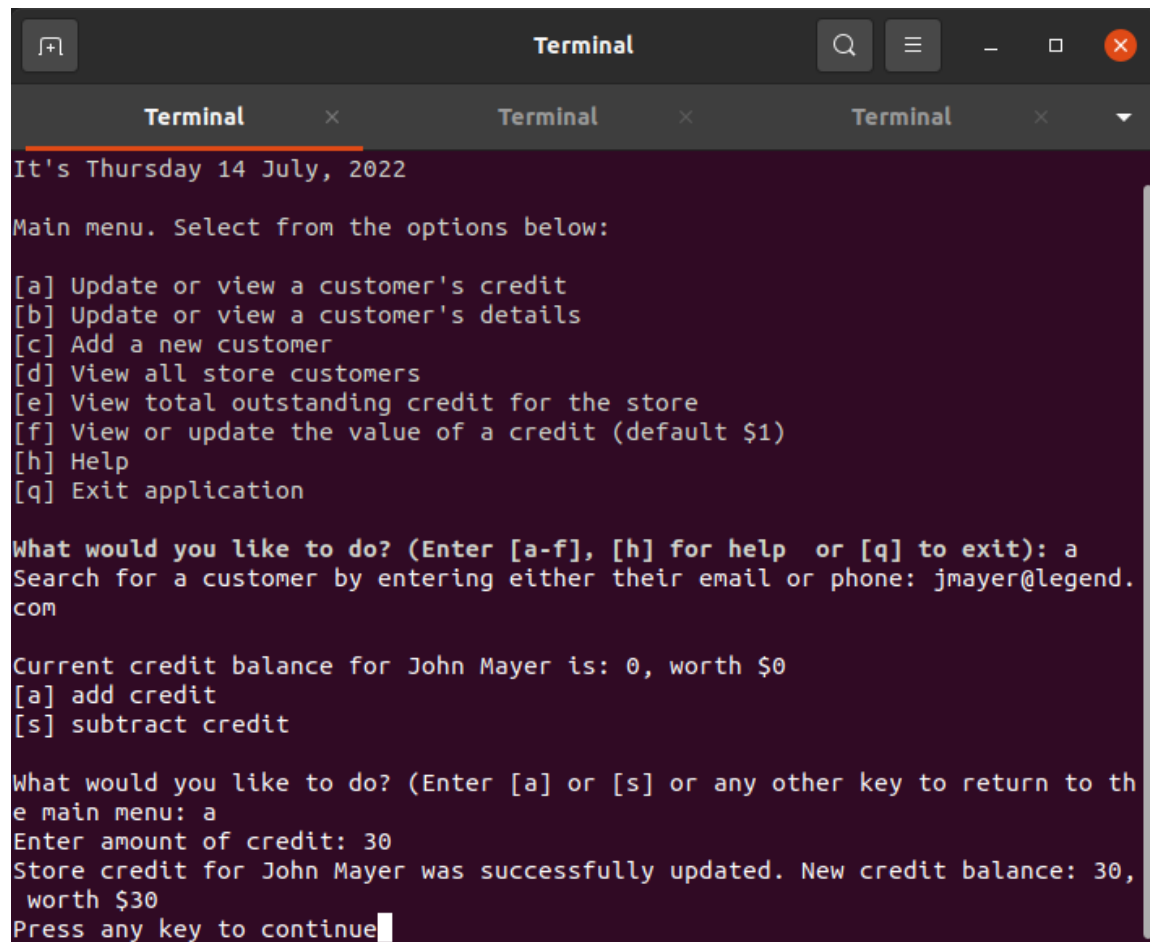
Current credit balance for John Mayer is: 0, worth $0
[a] add credit
[s] subtract credit

What would you like to do? (Enter [a] or [s] or any other key to return to the main menu): a
Enter amount of credit: 30
```

Application Walk-through

Update message displayed

Selecting option [a] indicates the credit will be added against a customer's name:

A screenshot of a macOS Terminal window with a dark theme. The window title is "Terminal". It shows the output of a program. The text is as follows:

```
It's Thursday 14 July, 2022

Main menu. Select from the options below:

[a] Update or view a customer's credit
[b] Update or view a customer's details
[c] Add a new customer
[d] View all store customers
[e] View total outstanding credit for the store
[f] View or update the value of a credit (default $1)
[h] Help
[q] Exit application

What would you like to do? (Enter [a-f], [h] for help or [q] to exit): a
Search for a customer by entering either their email or phone: jmayer@legend.com

Current credit balance for John Mayer is: 0, worth $0
[a] add credit
[s] subtract credit

What would you like to do? (Enter [a] or [s] or any other key to return to the main menu: a
Enter amount of credit: 30
Store credit for John Mayer was successfully updated. New credit balance: 30, worth $30
Press any key to continue
```

App Demo

Terminal App Logic and Code

The App works on three main layers:

- The main interface
 - Built using a while loop
- Two classes (Customer) and (Customers) class
 - The Customer class provides the attributes and methods for an individual customer, while the Customers class provides methods for operations on multiple customers.
- The linking functions that handle the logic between the interface and the class attributes and methods. Error handling used in this layer.

Terminal App Logic and Code

```
> __pycache__
```

```
customer.py
```

```
customers.py
```

```
help.py
```

```
main.py
```

```
mainfunct.py M
```

```
seed.py
```

```
$ startapp.sh
```

```
from customer import Customer

class Customers:
    # Customers class with list of of all customers (as a list) and
    def __init__(self, customer_list):
        self.customer_list = customer_list

    def show_customers(self):
        print("List of customers in the system")
        for customer in self.customer_list:
            customer.show_customer()

    def add_customer(self, fname, lname, phone, email):
        self.customer_list.append(Customer(fname, lname, phone, email))

    def find_customer(self, search):
        # iterate over customer list searching for customer
        # only look using phone or email as these are unique ids

        for customer in self.customer_list:
            if customer.phone == search or customer.email == search:
                return customer

    def delete_customer(self, customer):
        # using the find method above pass in the current customer
        self.customer_list.remove(customer)

        return print(f"{customer.firstname} {customer.lastname} was deleted")

    def update_customer(self, customer, update, new_value):
        if update == "first name":
            customer.firstname = new_value
        elif update == "last name":
            customer.lastname = new_value
        elif update == "phone":
```

The Development Process

- Since the application centres around one or more customers, I decided to use an OOP approach.
- I decided to also use modular approach to separate the code based on the relevant layer (interface, linking functions, classes).
- I decided to build the classes first and manually tested after each method or function was created.
- Used Trello to plan the development and implementation of each feature. (<https://trello.com/b/wpKDgCUi/terminal-application-implementation>)

Project Management

The screenshot displays a Trello workspace named 'Wade's Workspace' with a board titled 'Terminal Application Implementation'. The board contains five cards, each representing a feature of a 'Terminal Application'.

- Card 1: Terminal Application - Feature: Manage customer data**
Description: Add a new customer to the system or remove a customer from the system. And edit customer details.
Due date: 10 Jul, Progress: 9/9.
- Card 2: Terminal Application - Feature: manage customer credit**
Description: Add or delete store credit. Store credit can not drop below zero.
Due date: 13 Jul, Progress: 5/5.
- Card 3: Terminal Application - Feature: manage value of credit**
Description: Assign dollar value to store credit.
Due date: 14 Jul, Progress: 5/5.
- Card 4: Terminal Application - Feature: view total outstanding credit for the store**
Description: View total credit for the store.
Due date: 15 Jul, Progress: 0/5 + 15 Jul.
- Card 5: Terminal Application - Feature: Main menu**
Description: Main menu/interface.
Due date: 16 Jul, Progress: 0/5 + 15 Jul.

Each card includes a representative image (e.g., a person at a counter, a credit card, stacks of coins, US dollar bills, and a code editor) and a 'WD' (Wade) status icon. The interface also shows navigation options like 'Board', 'Workspaces', 'Recent', 'Starred', and 'Templates' at the top.

Challenges, Remaining tasks

- Main challenges around OOP, as this is relatively new to me.
- Favourite parts - whole thing!
- Foreseeable ethical issues include security for customer data. If the app were to be build with data persistence then user authentication etc. would be required to protect sensitive customer data.
- Still need to manually test the overall app and record the results.
- I would like to add a credit type category in the future.
- I would like to learn more about TDD and apply this in the future.