Wade Doolan
T1A3 - Terminal App Presentation

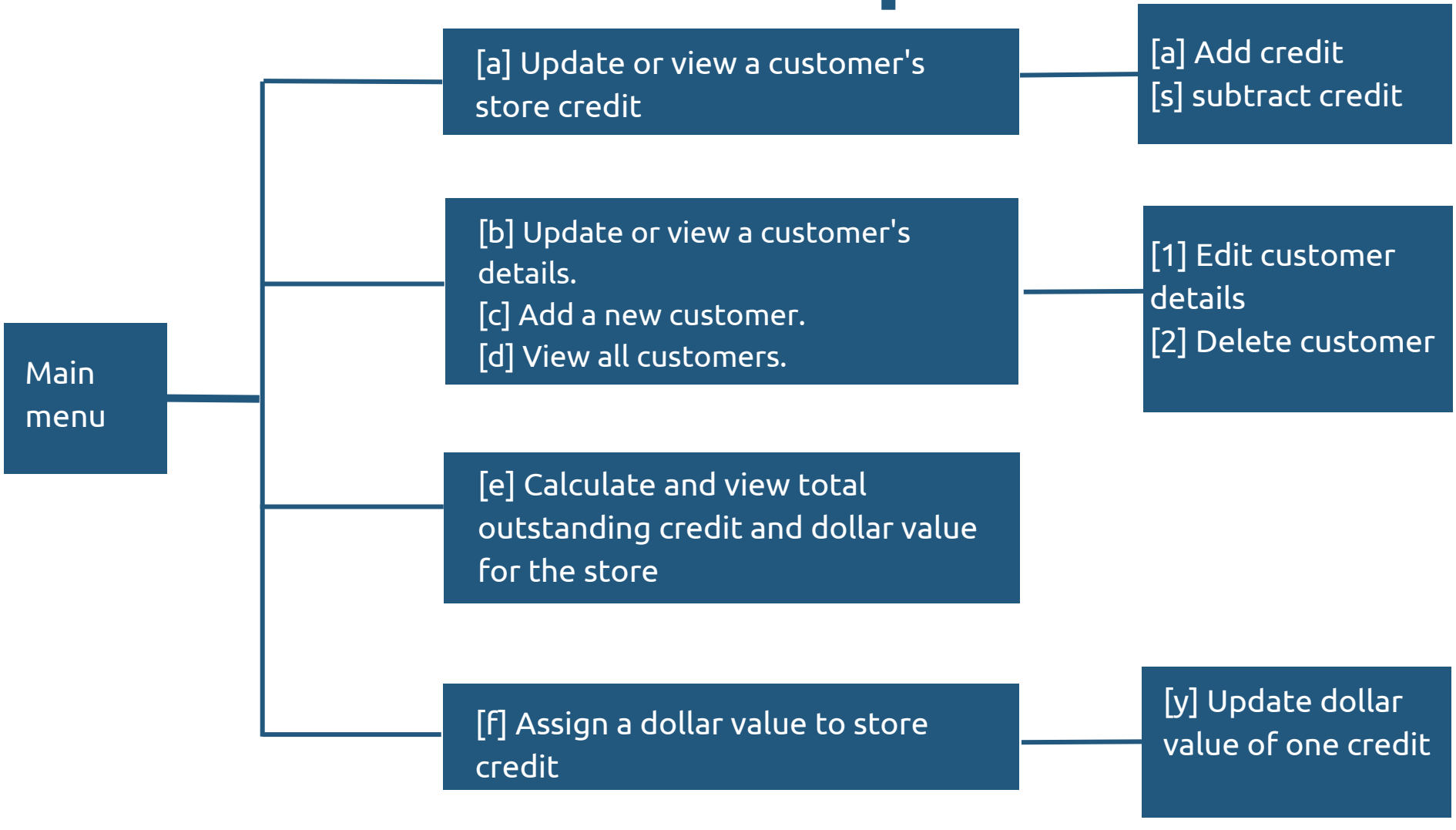
The Store Credit App

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.

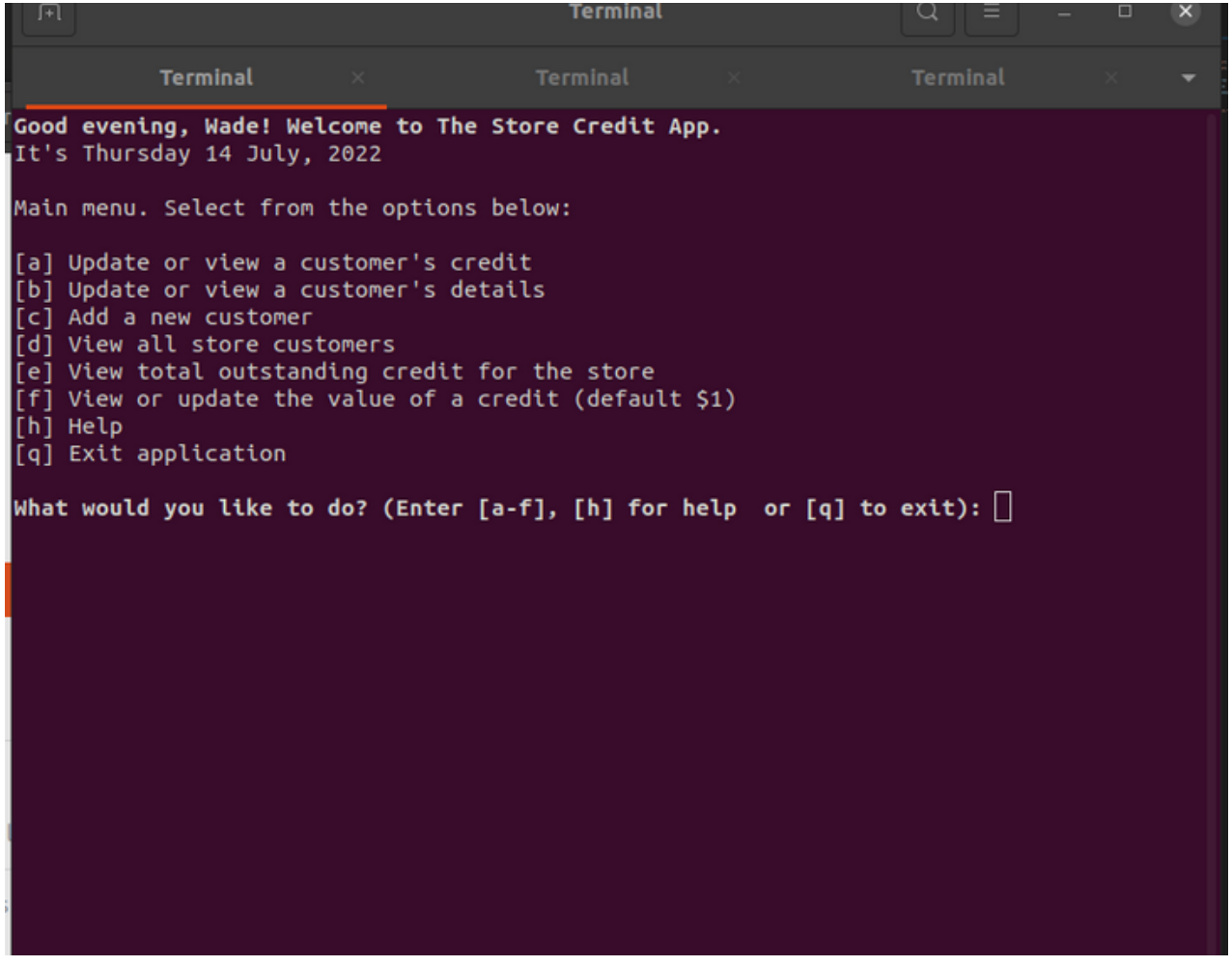
SiteMap


Main menu
[a] Update or view a customer's store credit
[a] Add credit
[s] subtract credit
[b] Update or view a customer's details.
[c] Add a new customer.
[d] View all customers.
[1] Edit customer details
[2] Delete customer
[e] Calculate and view total outstanding credit and dollar value for the store
[f] Assign a dollar value to store credit
[y] Update dollar value of one credit

Application Walk-through

Main Menu

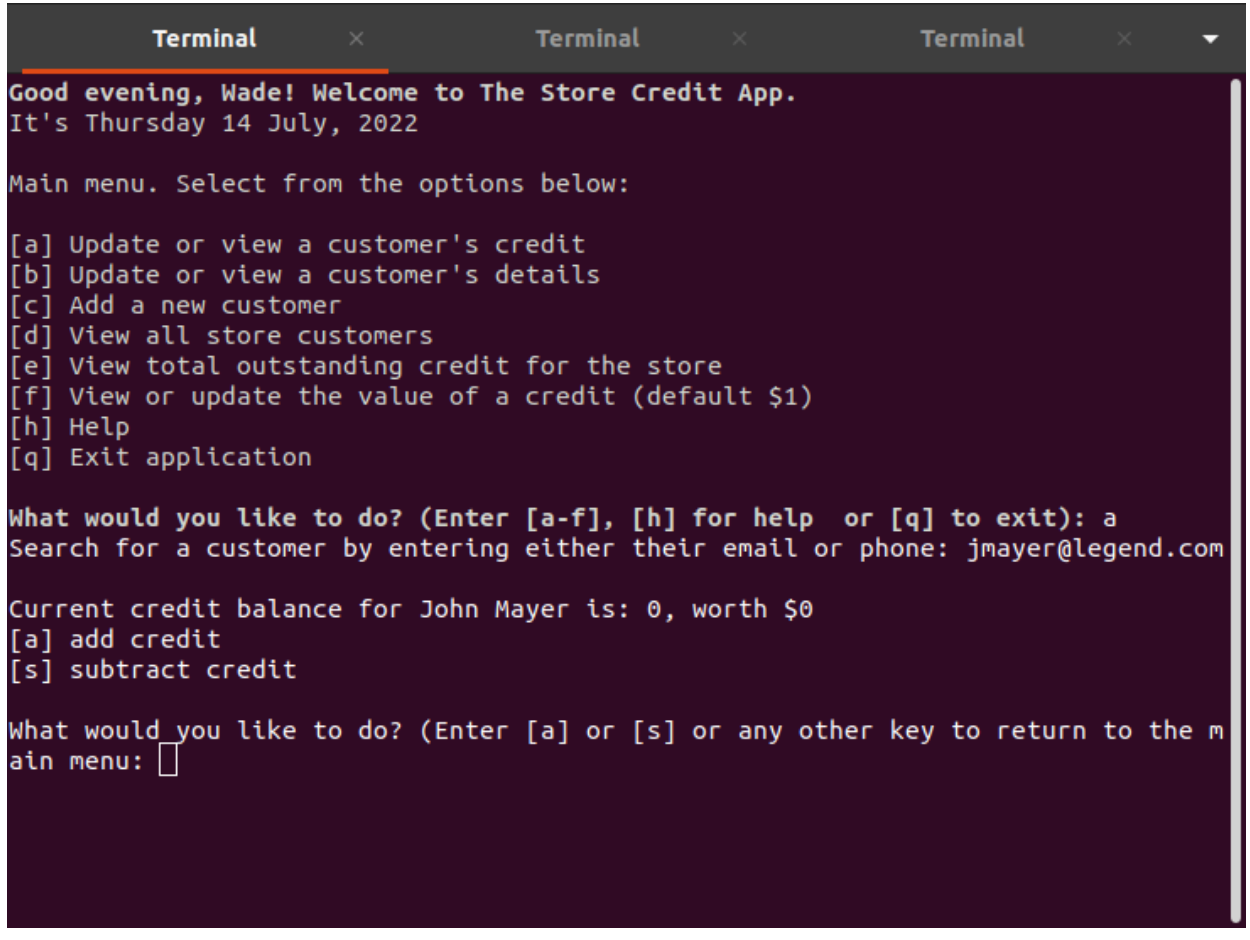
The application works around a main menu, which allows a user to carry out various tasks. Menu shown below:



Application Walk-through

View/update customer credit

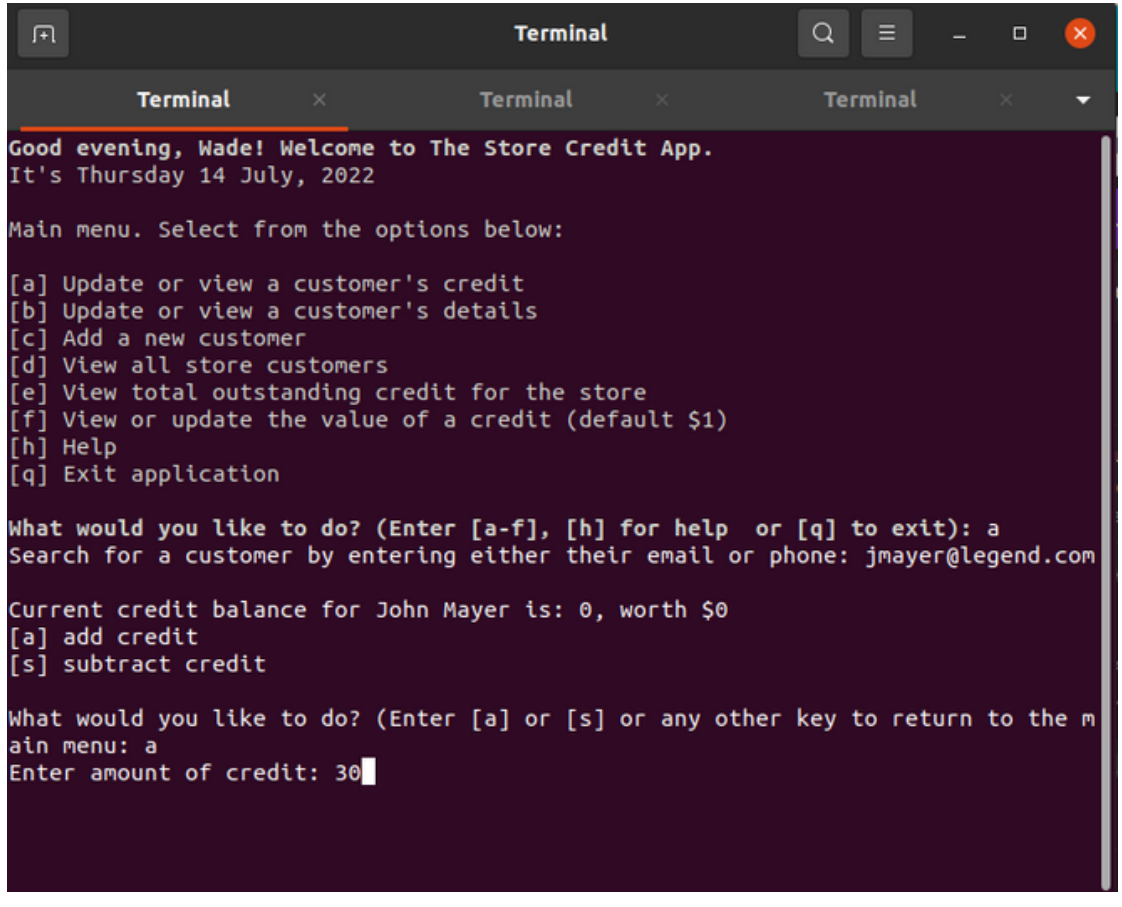
Selecting option [a] first prompts the user to search for a customer. When the customer is found the following menu is displayed:



Application Walk-through

Adding customer credit

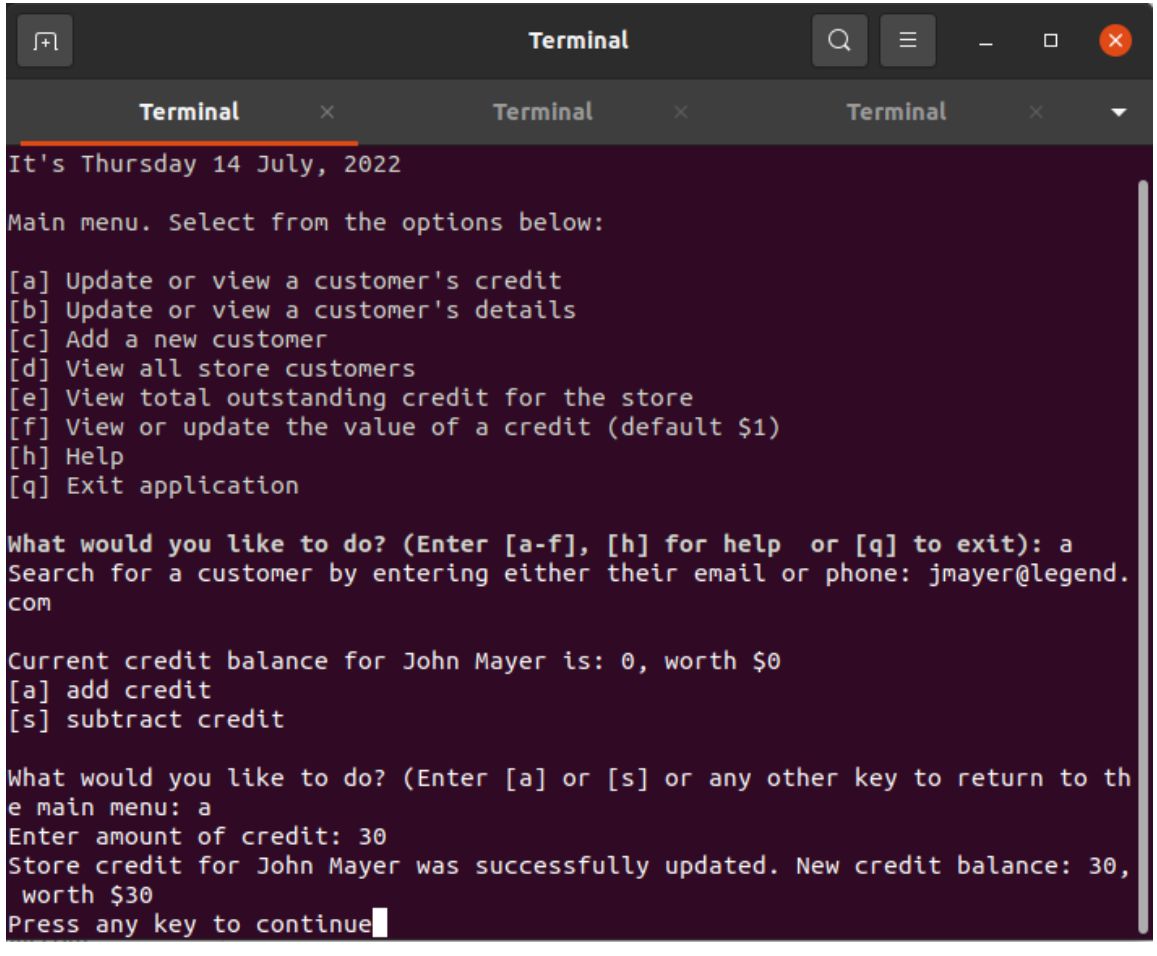
Selecting option [a] indicates the credit will be added against a customer's name. The user then enters the amount of credit to be added.



Application Walk-through

Update message displayed

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



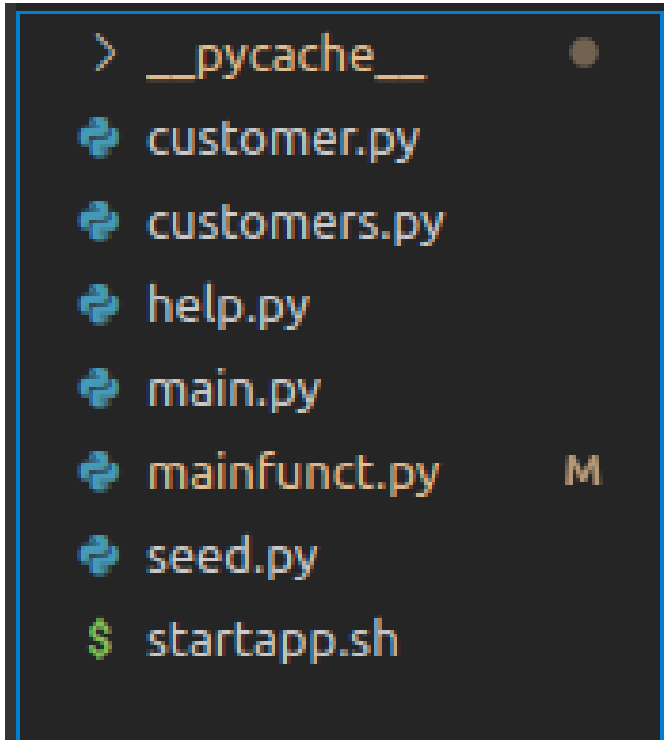
App Demo

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

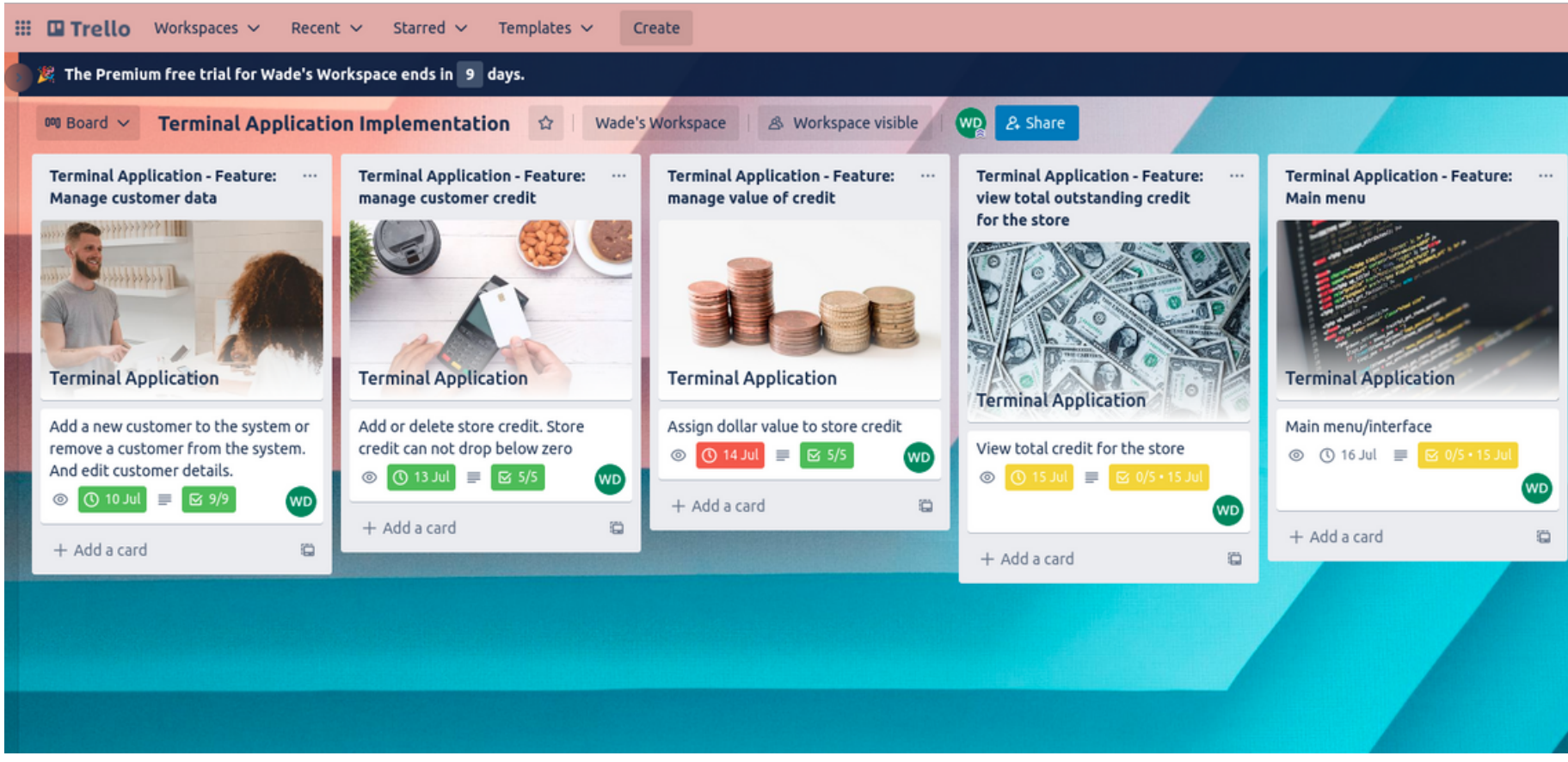



```python
from customer import Customer


class Customers:
    # Customers class with list of of all customers (as a list) ar
    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, em

    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 == searc
                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} wa

    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



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.