Group 26 - Assignment 2 Design Document

Jack Caldarone, Nicholas Tillo, Daniel Kamenetsky

Overview

For this prototype, we chose to store all our information in comma separated value (csv) files. Because of this, most of our functions write to or read from csv files, rather than creating classes and editing classes. The reason for this is because we wanted to simulate how the actual software would work, and without writing our data to a separate file there would be no way to save the data we had entered. There is one exception however, which is the cart class. Since the cart is filled with data specific to a user, it does not need to be stored in a database along with all other info on restaurants, users, items, etc.

The following functions were designed to work with the cart class:

get_total_price()

- Class function that takes no input. Sums up the total price of every item currently in the cart, and then prints the total.

add_to_cart()

- Takes an item from a restaurant's menu as input, and adds it to the cart.

remove_from_cart()

- Takes an item from a restaurant's menu as input, and removes it from the cart.

check_cart()

- Displays all the items in the cart and calls the get_total_price() function.

For user registration/ login, there are the following functions:

register_user()

- Takes user inputted parameters (username, password, email, location, role) as input. Checks to see if any other users have the inputted usernames. If the username is original, a new line is written to the csv file that stores pre-existing users.

login_user() -

- Takes user inputted username and password as parameters. Checks csv file storing login information to verify that the information is correct. Gives either a success or failure message afterwards.

For delivery and payment, there are the following functions:

get_card_info()

- Gets the user to input their credit card information (number, CVV, expiration date)

create_delivery_status()

- Takes the total cost of all items in the user's cart as input and writes the price, order id, and item list to the delivery csv file.

check_delivery_status()

- Takes an order id as input. Checks the delivery csv file for its current status, which is then outputted.

For viewing restaurants/menus, there are the following functions:

select_restaurant_and_view_menu() -

- Takes the user location as input. Matches this information with the restaurants csv file, to check which restaurants are in their area. Calls other smaller functions mainly to use tkinter to display the correct information.

For restaurant/menu management, there are the following functions:

create_restaurant()

- Checks csv file if there is a restaurant already made by this user. If the user hasn't already made a restaurant, the user is asked for input on restaurant details. Adds this information to the csv file.

delete_restaurant()

- Checks csv file to see if the user owns a restaurant. If the user does, their restaurant is removed from the csv file.

edit_restaurant()

- Checks csv file to see if the user owns a restaurant. If the user does, they are prompted on which attribute they want to change, and what they want to change it to.

create_item()

- Checks csv file to see if the user owns a restaurant. If the user does, they are prompted on the attributes of the item they wish to add. The new item is then created, and added to the csv file.

remove_item()

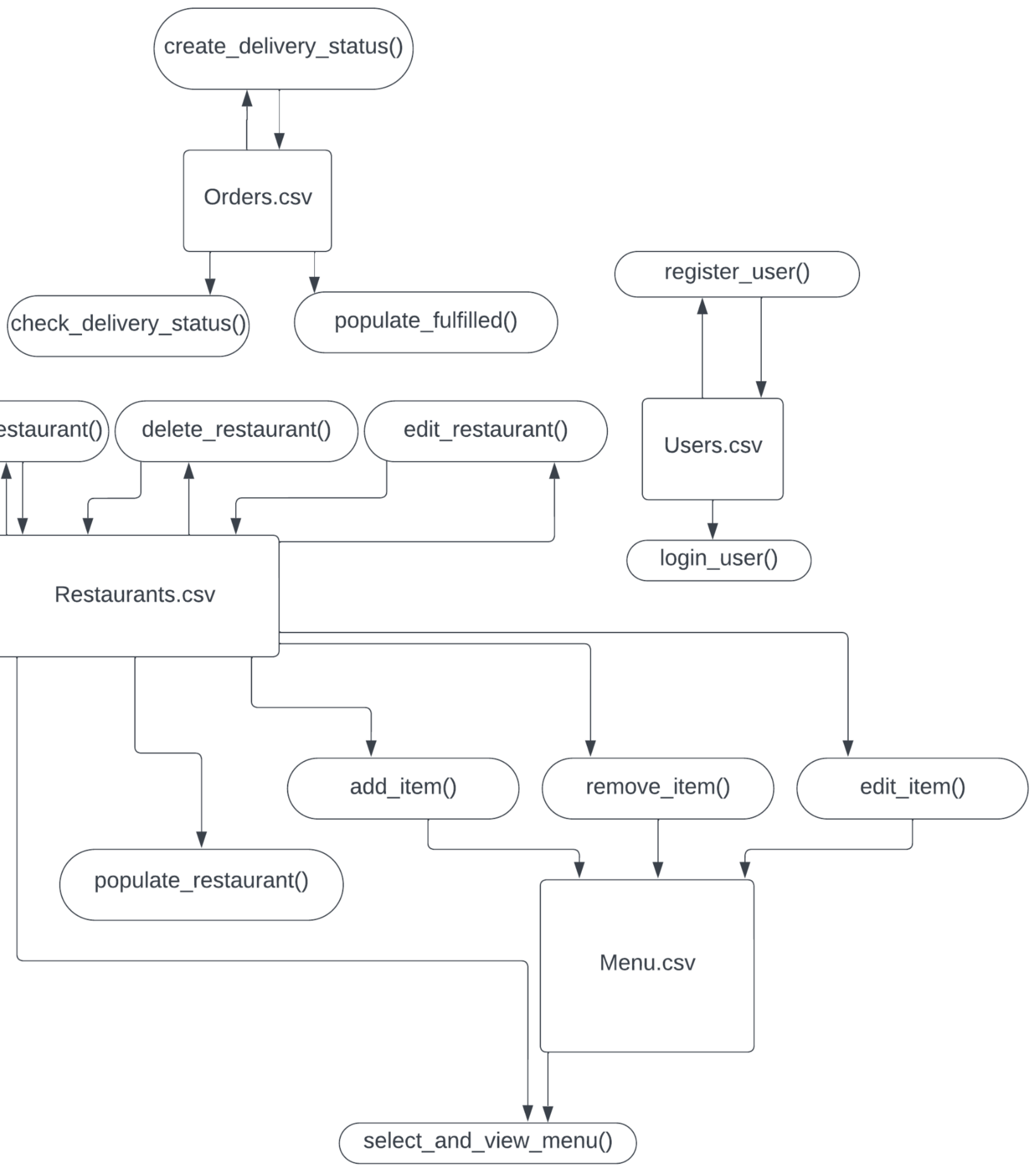
- Checks csv file to see if the user owns a restaurant. Asks the user what item they want to delete. Checks csv file to see if the item exists, if it does it is removed.

edit_item()

- Checks csv file to see if the user owns a restaurant. If the user does, they are prompted on the attribute of the item they wish to change. Check the csv file to see if the item they wish to edit exists. If it does, the selected attribute is changed.

Diagram of Architecture

Below is a detailed flowchart of how the functions interact with the spreadsheets. A box indicates an item is a spreadsheet, and an oval indicates something is a function. An arrow leading towards a box indicates that the function writes to the spreadsheet, and an arrow leading towards a function indicates that a function reads from a spreadsheet


create_delivery_status()
Orders.csv
check_delivery_status()
populate_fulfilled()
register_user()
Users.csv
login_user()
create_restaurant()
delete_restaurant()
edit_restaurant()
Restaurants.csv
add_item()
remove_item()
edit_item()
populate_restaurant()
Menu.csv
select_and_view_menu()