Gustavo Jimenez T1A3 - Self Service Kiosk

Link to video presentation

GustavoJimenezT1A3_VideoPresentation

R1. Answers to all the documentation requirements below.

R2. Your README.md should have a separate heading for each documentation requirement and answers organised under the appropriate headings.

R3. Provide full attribution to referenced sources (where applicable).

rich library ---- https://www.youtube.com/watch?v=4zbehnz-8QU
https://rich.readthedocs.io/en/latest/index.html

alive-progress library --- https://pypi.org/project/alive-progress/ https://pypi.org/project/alive-progress/

json files --- https://www.pythoncheatsheet.org/cheatsheet/json-yaml

pytest --- https://www.youtube.com/watch?v=ucjRpS7WCPA

R4. Provide a link to your source control repository

- GustavoJimenez_T1A3-GitProject

R5. Identify any code style guide or styling conventions that the application will adhere to.

Reference the chosen style guide appropriately.

The code is well-organized and follows PEP 8 and PEP 257 conventions. Here are a few points I kept in mind:

- Imports: Kept them at the top, but re-organised and grouped appropriately.
- Whitespace: use of whitespace is maintained around operators and after commas.
- Indentation: The code is consistently indented using 4 spaces per level.
- String Formatting: F-strings are used for string formatting, which is more concise and readable.
    - Some string were not F-string for personal readability
- Comments/Docstrings: Applied informative comments, I used triple double-quotes.
- Function Documentation: Functions have docstrings describing their purpose, parameters, and return values.
- Variable Names: Variable names are descriptive, and there are no formats inconsistencies.
- Consistent Case format: Snake case for both variable and function names.

R6. Develop a list of features that will be included in the application. It must include:

- at least THREE features
- describe each feature

  Note: Ensure that your features above allow you to demonstrate your understanding of the following language elements and concepts:

  - use of variables and the concept of variable scope
  - loops and conditional control structures
  - error handling

Consult with your educator to check your features are sufficient .

User-friendly interface:

The interface should be easy to navigate and understand, with clear instructions and options for the user. Displaying the menu in a clear and organised manner, with categories and prices for each item.

Create user registration, user login, login password recovery, and continue as visitor

- Allowing and encouraging users to register to have access to special products on the menu and other perks.
- Password and login to access the specials and have a personal database
- Password recovery, if the user forgot the code, this can be retrieved by providing name and birthday.
- A simple continue as visitor or first time buyer

User Order tracking:

Providing users with personal database to provide special products, customised orders, provide discounts.

Order display and remove items:

Allowing users to add and preview the order. Remove unwanted items, preview and continue with purchase

Order Finalization:

After the user has reviewed their order, the kiosk should confirm the order and display a message with an estimated wait time. It should also provide a receipt with the order details and total cost.

Order time to prepare:

Allowing users to see the time it will take and a progress bar indicating when the meal would be ready

Note: Ensure that your features above allow you to demonstrate your understanding of the following language elements and concepts: - use of variables and the concept of variable scope - loops and conditional control structures - error handling

- Consult with your educator to check your features are sufficient.

My Wireframe/Diagram

R7. Develop an implementation plan which:

Set Up Project Structure:

Create a new Python project directory.
Set up version control (e.g., Git) for your project.
Create Initial Files:

- Create a main Python script (e.g., ordering_kiosk.py) to serve as the entry point for your application.
- Create a menu file (e.g., menu.json) to store your menu items.

Implement Menu Display

- Read the menu data from menu.json in your main script.
- Implement a function to display the menu with item numbers, categories, and prices.
- Test the menu display functionality.

Implement Order Placement:

- Implement a function to handle order placement, allowing users to add items to their order by entering item numbers and quantities.
- Test order placement functionality.

Implement Optional Features / Test / Debug

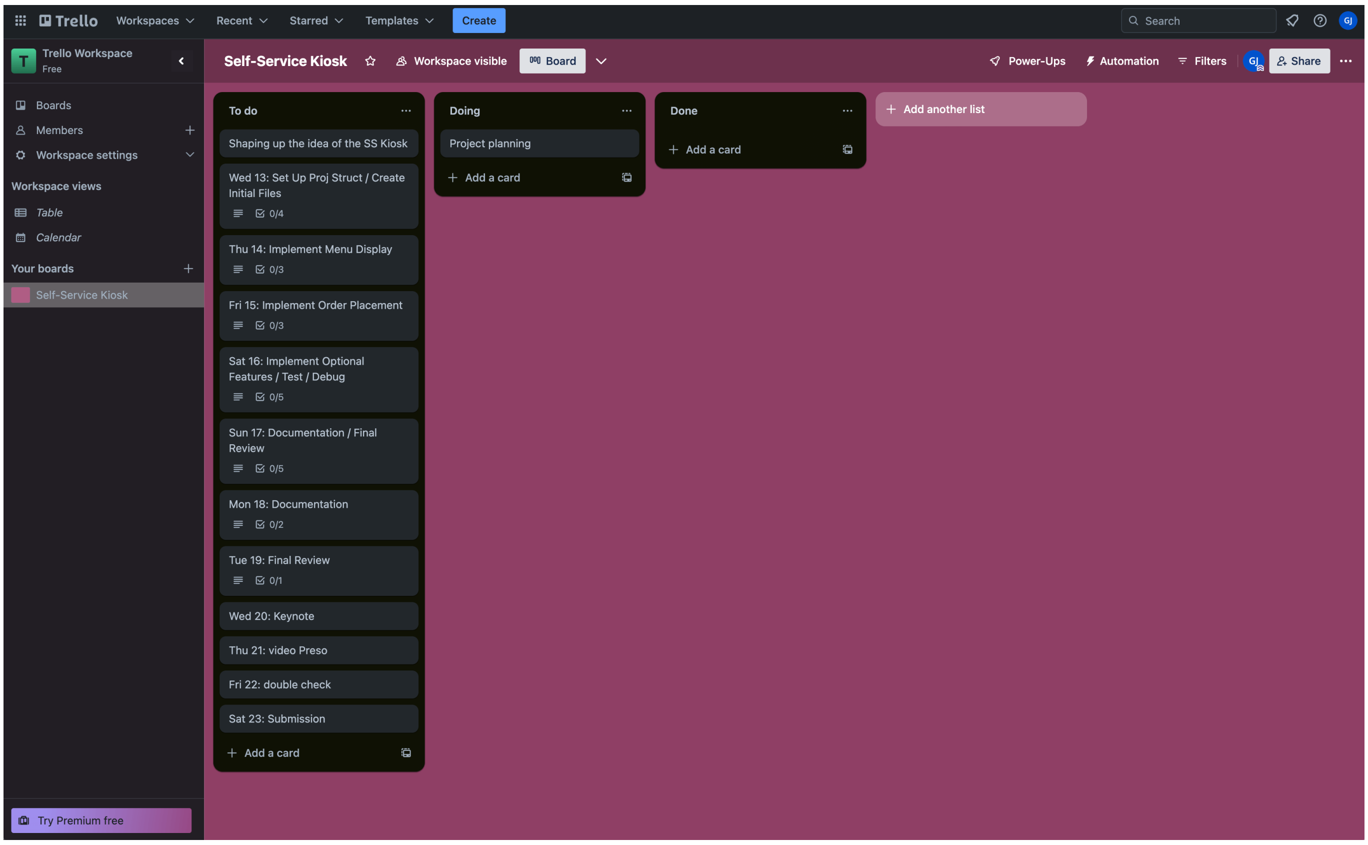
1. Implement Order Summary:
   - Create a data structure to store the user's order.
   - Implement a function to display a summary of the user's order, including item names, quantities, and total cost.
   - Test the order summary functionality.
2. Implement Order Finalization:
   - Implement a function to confirm the order, display an estimated wait time, and provide a receipt with order details and total cost.
   - Test the order finalization functionality.

Implement Error handling:
Implement error handling for invalid input and items. - Register page - user register error handling - Register page - user login error handling and 3-try-limit - Register page - continue as a visitor error handling

---

Link to my project in Trello

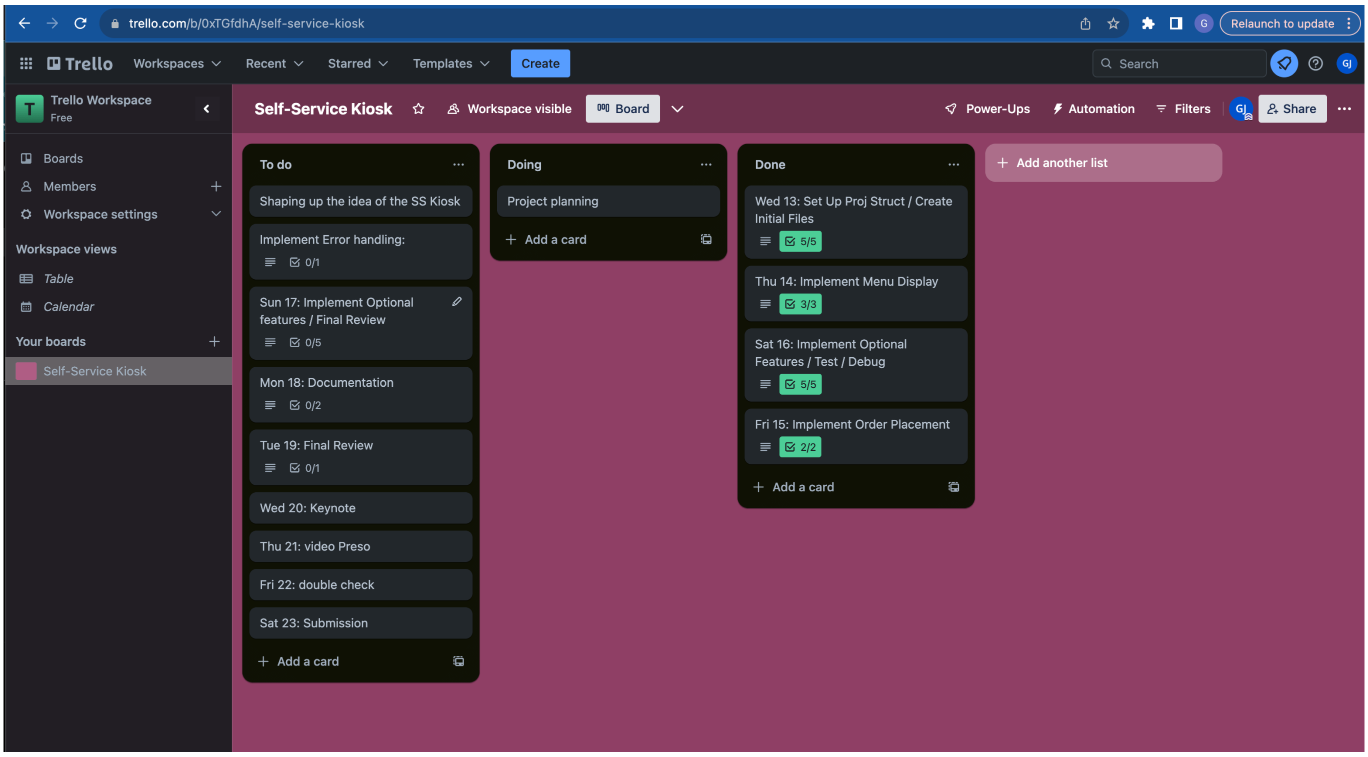
Trello progress



To do
- Shaping up the idea of the SS Kiosk
- Wed 13: Set Up Proj Struct / Create Initial Files 0/4
- Thu 14: Implement Menu Display 0/3
- Fri 15: Implement Order Placement 0/3
- Sat 16: Implement Optional Features / Test / Debug 0/5
- Sun 17: Documentation / Final Review 0/5
- Mon 18: Documentation 0/2
- Tue 19: Final Review 0/1
- Wed 20: Keynote
- Thu 21: video Preso
- Fri 22: double check
- Sat 23: Submission
- Add a card

Doing
- Project planning
- Add a card

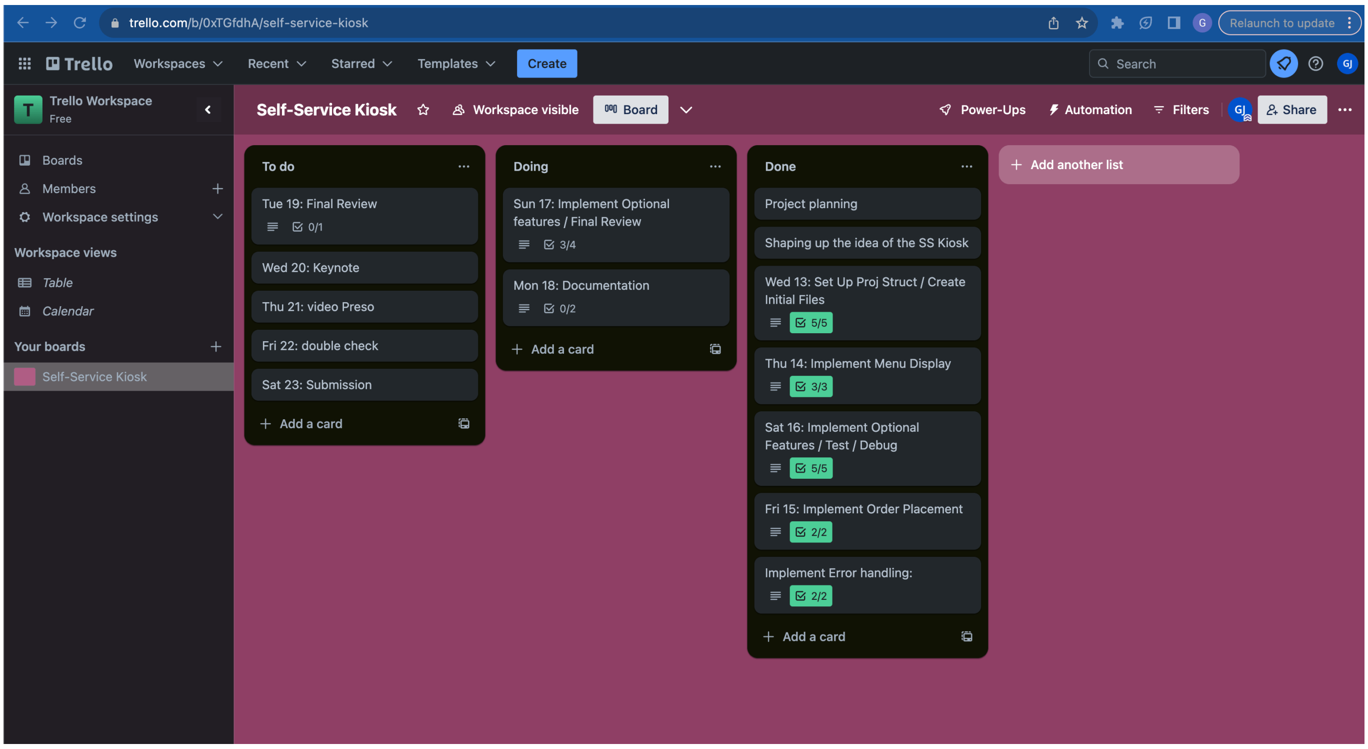
Done
- Add a card



To do
- Shaping up the idea of the SS Kiosk
- Implement Error handling: 0/1
- Sun 17: Implement Optional features / Final Review 0/5
- Mon 18: Documentation 0/2
- Tue 19: Final Review 0/1
- Wed 20: Keynote
- Thu 21: video Preso
- Fri 22: double check
- Sat 23: Submission
- Add a card

Doing
- Project planning
- Add a card

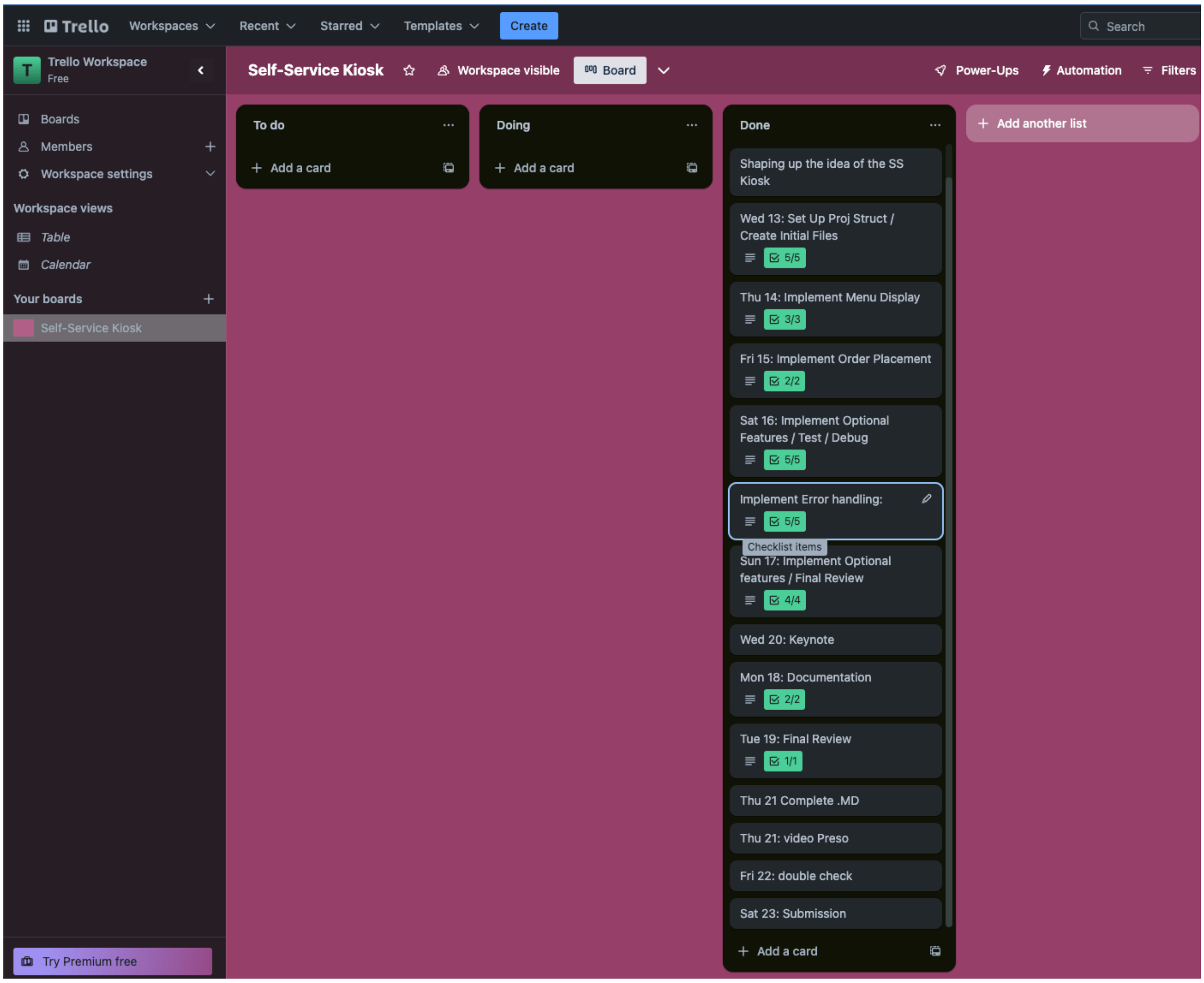
Done
- Wed 13: Set Up Proj Struct / Create Initial Files 5/5
- Thu 14: Implement Menu Display 3/3
- Sat 16: Implement Optional Features / Test / Debug 5/5
- Fri 15: Implement Order Placement 2/2
- Add a card



To do
- Tue 19: Final Review — 0/1
- Wed 20: Keynote
- Thu 21: video Preso
- Fri 22: double check
- Sat 23: Submission

Doing
- Sun 17: Implement Optional features / Final Review — 3/4
- Mon 18: Documentation — 0/2

Done
- Project planning
- Shaping up the idea of the SS Kiosk
- Wed 13: Set Up Proj Struct / Create Initial Files — 5/5
- Thu 14: Implement Menu Display — 3/3
- Sat 16: Implement Optional Features / Test / Debug — 5/5
- Fri 15: Implement Order Placement — 2/2
- Implement Error handling: — 2/2



To do

Doing

Done
- Shaping up the idea of the SS Kiosk
- Wed 13: Set Up Proj Struct / Create Initial Files — 5/5
- Thu 14: Implement Menu Display — 3/3
- Fri 15: Implement Order Placement — 2/2
- Sat 16: Implement Optional Features / Test / Debug — 5/5
- Implement Error handling: — 5/5
- Sun 17: Implement Optional features / Final Review — 4/4
- Wed 20: Keynote
- Mon 18: Documentation — 2/2
- Tue 19: Final Review — 1/1
- Thu 21 Complete .MD
- Thu 21: video Preso
- Fri 22: double check
- Sat 23: Submission

R8. Design help documentation which includes a set of instructions which accurately describe how to use and install the application.

A file called SS_Kiosk_User_Documentation.MD has been created to help the user operating the terminal application

---