**PySquad Proposal**

**Team Members:**

* Zachary Herald
* Mark Klein
* Krystal Michaelis
* Bill Parrish
* Orlando Valadez
* Jared Volpenhein

**Client:**

* Local restaurant (or any specific type of restaurant if needed)

**System Scope:**

The restaurant ordering system will allow customers to place orders, view available menu items, and manage orders seamlessly from a digital interface. It will feature:

* **Menu Display:** A list of food and drinks with prices.
* **Order Creation:** Customers can add items to their cart, view the cart, and place the order.
* **Order Tracking:** The restaurant can track orders in real-time.
* **Customer Interaction:** Customers can update or cancel orders, with the system reflecting those changes immediately.

**Purpose of the System:**

* **Simplifying the Ordering Process:** Customers can quickly browse the menu, customize their order, and place it digitally.
* **Efficient Order Management:** The restaurant staff can easily track, update, and manage orders.
* **Improved User Experience:** A clean and responsive graphical interface for ease of use, making the ordering process smoother for both the customer and the restaurant.

**System Features:**

* **Graphical User Interface (GUI):**
  + A user-friendly interface where customers can interact with the system.
  + Displaying available menu items (using Python libraries like Tkinter or PyQt for GUI).
* **Backend Structure:**
  + Utilize three classes (e.g., Menu Item, Order, Customer) to manage the data and functionality of the system.
  + Use collections like lists, tuples, or dictionaries to organize menu items, orders, and customer data.
* **Order Management:**
  + Track items added to an order, the total cost, and order status (e.g., pending, completed).
  + Allow customers to cancel or modify their orders before final submission.
* **Error-free Execution:**
  + Ensure that the system runs with no syntax or runtime errors.

**Team Roles:**

1. **Project Leader:** Krystal Michaelis
   * Oversee the overall project, manage timelines, and ensure the team adheres to the goals. Facilitates team communication and project organization using tools like GitHub and Kanban boards. Manages the GitHub repository, making sure all code is shared appropriately, and the project is structured well. They will also facilitate the team’s weekly meetings and ensure proper communication.
2. **Backend Developer(s):** (two developers)
   * Responsible for the logic and functionality of the system. They will create classes and implement functions that handle orders, menu items, and customer interactions.
3. **Frontend Developer(s):** (two developers)
   * Responsible for designing and developing the user interface. They will create the GUI layout and ensure the system is user-friendly.
4. **Testing & Documentation Specialist:** (one tester)
   * Ensure the system is tested thoroughly. They will document the process and results, ensuring the system is free of bugs and the user manual is complete.

**Tools and Technology:**

* **Python**: The primary programming language for the system.
* **Tkinter or PyQt**: For the graphical user interface.
* **GitHub**: For version control and collaborative code management.
* **Kanban Boards (Trello, Jira)**: To track progress and manage tasks.
* **Python Collections (lists, dictionaries, etc.)**: To store data about menu items, orders, and customers.

**Timeline & Milestones:**

* **Week 1:**
  + Team introduction, roles assignment, initial project planning, and setting up GitHub repository and Kanban board.
* **Week 2-3:**
  + Backend development (creating classes like Menu Item, Order, and Customer).
  + Frontend development (UI design).
* **Week 4-5:**
  + Integration of backend and frontend.
  + Initial testing of functionalities.
* **Week 6:**
  + Implement features like order tracking and customer interaction.
  + Continue refining UI.
* **Week 7:**
  + Testing and bug fixing.
  + Prepare final documentation (class diagrams, report of results).
* **Week 8:**
  + Final submission, project evaluation, and reflection.

**Communication Plan:**

* Weekly team meetings via Zoom to check progress, discuss challenges, and reassign tasks if needed.
* Use Group Messaging for daily communication.
* Use GitHub for sharing code, tracking issues, and making contributions visible to everyone.