Project Name: Kitchen Story **Date:25 July 2023**

Project Objective: To create an e-commerce portal that lets people shop basic food items on their website.

Developer: Teboho Innocent Mosiea

GitHub Link: *https://github.com/teboho46664/Kitchen_Story.git*

Features:

A. Search form in the home page to allow entry of the food items to be purchased by the customer.
B. Based on item details entered, it will show available food items with price.
C. Once a person selects an item to purchase, they will be redirected to the list of available items. In the next page, they are shown the complete breakout of the order and details of the payment to be made in the payment gateway. When payment is done, they are shown a confirmation page with details of the order.
D. Admin login page where admin can change password after login if he wants to.
E. A master list of food items available for purchase.
F. A functionality to add or remove food items.

User Stories:

1. As a customer, I want to be able to search for food items that I want to purchase.
2. As a customer, I want to be able to see a list of available food items with prices.
3. As a customer, I want to be able to add items to my cart and checkout.
4. As a customer, I want to be able to pay for my order using a secure payment gateway.
5. As an admin, I want to be able to login to the website and manage the food items.
6. As an admin, I want to be able to add or remove food items from the website.

Acceptance Criteria:

- ✓ The website should be able to search for food items by name, category, or price.
- ✓ The website should be able to display a list of available food items with prices.
- ✓ The website should be able to add items to the cart and checkout.
- ✓ The website should be able to process payments through a secure payment gateway.
- ✓ The admin login page should be secure and only accessible to authorized users.
- ✓ The admin should be able to add or remove food items from the website.

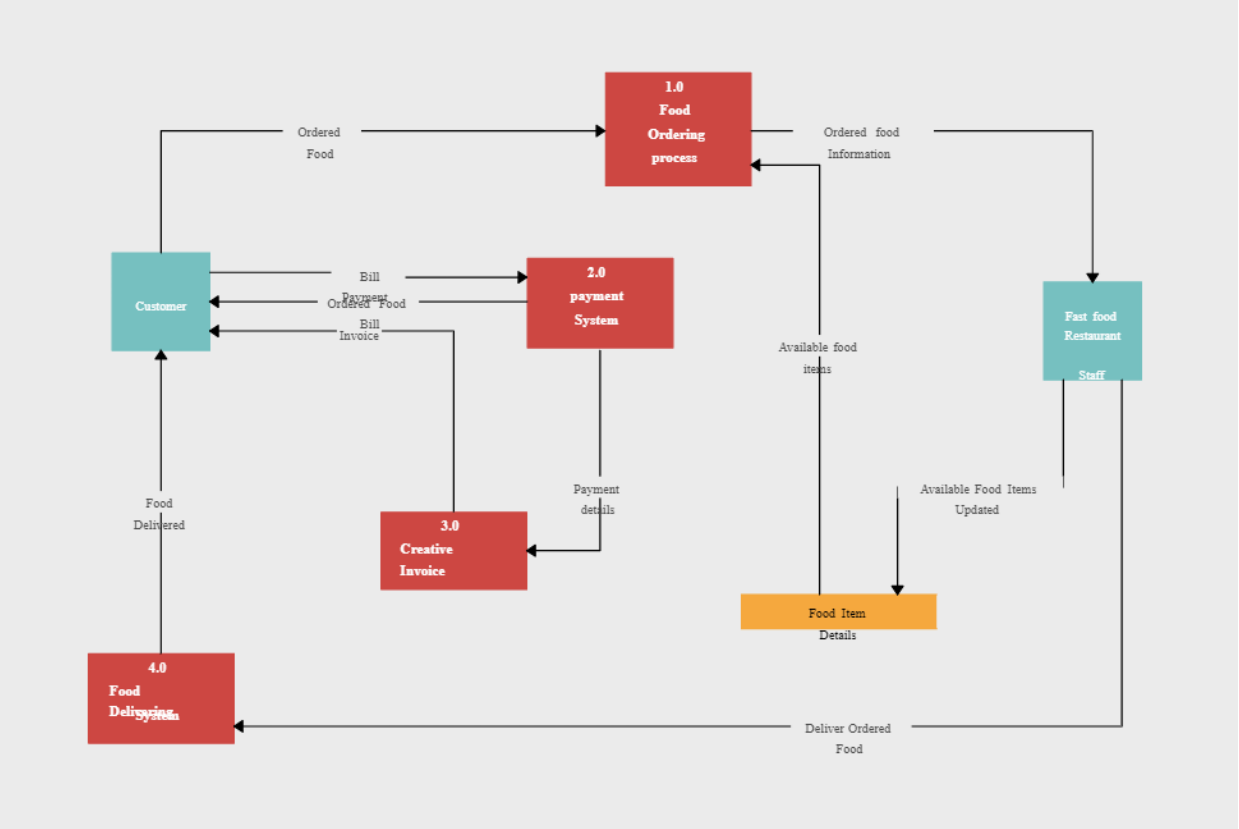
Testing Plan:

- The website will be tested by a team of testers.
- The testers will use a variety of test cases to ensure that the website meets all the acceptance criteria.
- The testers will report any bugs or defects to the development team.
- The development team will fix any bugs or defects reported by the testers.

Deployment Plan:

- The website will be deployed to a staging environment for testing.
- Once the website has been tested and approved, it will be deployed to the production environment.
- The website will be monitored for any performance or security issues.

Flow Diagram



Maintenance Plan:

- The website will be maintained by a team of developers.
- The developers will fix any bugs or defects reported by users.
- The developers will also add new features to the website as needed.

Sprint Planning:

Sprint 1:

- Create the search form and functionality.
- Create the list of available food items.
- Create the checkout functionality.

Sprint 2:

- Integrate the payment gateway.
- Create the admin login page.
- Add the functionality to add or remove food items.

Sprint 3:

- ✓ Test the website and fix any bugs.
- ✓ Deploy the website to the production environment.
- ✓ Possible Future Upgrades and Development:
- ✓ Add a blog section to the website where users can share recipes and cooking tips.
- ✓ Add a forum where users can discuss food and cooking.
- ✓ Add a social media integration so that users can share their purchases on their social media profiles.
- ✓ Add a loyalty program so that users can earn points for their purchases and redeem them for discounts or free items.
- ✓ Develop a mobile app for the website so that users can shop from their smartphones and tablets.
- ✓ Add tracking of parcel or delivery of food items so that users can track their orders and see when they will be delivered.
- ✓

Core concepts used in the project:

- Object-Oriented: used to create and model objects for users and their credentials.
- REST API: used to communicate between the back-end and the front-end sides.
- Data Access Object: to abstract and encapsulate all access to the data source.
- Object–Relational Mapping: to map the objects to the database.
- Databases: used to store and retrieve data.
- Data Sources: used to define a set of properties required to identify and access the database.
- Collections: used some collections such Arraylist to store collection of data.
- Exception Handling: used to catch problems that arises in the code especially in I/O blocks.