1. **Introduction**

The name of this project is “Cashier Application for CRISTY’S LOVE BURGER HUB POS System”. CRISTY’S LOVE BURGER HUB is a fast food restaurant that specializes in burgers for the patrons.

This application is developed for streamlining the order and the transaction process between the cashier and the customer. The prioritisation of this software is the swiftness, simplicity and efficiency in order-taking. The system, unlike other available applications available in the market, is PIN protected and can only be accessed by the manager at first and then subsequently the assigned cashier can log into his/her respective account. This double authentication ensures null mishandling of the system itself and restaurant assets (viz. embezzlement of counter-cash). A successful login provides a user friendly and an ordered user interface for order input along with order status, product details and product delivery options.

This system functions under CRUD application. The database stores restaurant staff’s credentials for login authentication purpose, product details for customer’s purpose, and order details for delivery purpose as well as for restaurant record purpose. The recorded data is retrievable and mutable. It also supports registration of frequent customers for membership to enjoy regular privileges.

1. **Aim**

The aim of this counter cashier application is to take customer’s orders efficiently in a systematic manner to curb the time taken in the process ensuring less queue time for other customers, instant notification of orders to the production department in the kitchen and swift delivery to the consumers.

1. **Objectives**

The objectives of the cashier software are as follows:

1. Ensures security of the staff account and restaurant assets.
2. User friendly interface facilitates easiness to operate the system.
3. Employee login protocol secures the system.
4. Analyses sales by salesperson or by product.
5. Inputs via touchscreen thus eliminating use of mouse and keyboard.
6. Track sales, taxes, employee performance and customer frequency.
7. Facilitates combi-orders, and accuracy in calculations of discounts compared to single order.
8. Create better and less hectic working environment for the staffs.
9. Replace the use of traditional pen and paper order taking system.
10. **Problem Statement**

The traditional system of noting customer’s order in a restaurant with pen and paper is not on par with the current technologically developing time. The data organisation and storage is a hassle and many transactions are inconsistent due to mishandling of cash. Since taking orders manually takes a lot of time, the kitchen staff starts production late and hence the guests receive their order also late. Therefore, there is sever lack of co-ordination among the staff-members. Many guests have to wait for their turn longer.

This project states these aforementioned problems by digitising the order process, which informs the kitchen-personnel instantly and hence saves time in both order input as well as production and warrants quick delivery of order. The mandatory login of the manager and then of the cashier handles misuse of company assets. The easy-to-use UI enables hassle-free and correct orders input. The system calculates correctly the total cost of the order and guarantee lesser to none transaction mistakes. The integrated database secures the history of orders and the product details.

1. **Features**

The features of this system are as follows:

1. Create and manage staff(s) profile, thus
2. Supports multiple user accounts as per need
3. Manager allows the login of staffs only after her permission, thereby
4. Doubling the security of login in the system
5. Simple and user friendly UI
6. Allows single order and combi-order without any calculation mistakes
7. Observable status of the orders
8. Easy modification of orders
9. Swiftness in order, billing and transactions, hence
10. Swiftness in production and delivery
11. **Functional Requirements**

The Functional requirements that define what a software must do, what its features and functions are, as follows:

1. Double security during signing in the software
2. Addition and modification of the staff and product database
3. Products in their respective category
4. Possibility of combi-menu and kids-menu
5. Options of takeaway or eat-in
6. Registration of regular customers for membership
7. Live update of order status
8. Digital calculation and billing system
9. **Non-Functional Requirements**

The non-functional requirements that explain how the system should perform are as follows:

1. The staff should be able to login within 10 seconds of entering his/her pincode
2. The touch button should input order instantly with maximum lag of 5 seconds
3. Modification of data in database should be quick.
4. Calculation of total amount should not take more than 10 seconds
5. Tab menus should work within 5 seconds.
6. The OTP for membership registration should be generated and forwarded within 20 second
7. Log out should take place within 10 seconds
8. **Scope**

The scope of this Point of Sale (POS) cashier system is defined collectively by all the preceding topics of aim, objectives, problem statements, features, functional requirements and non-functional requirements. The main scope of this program is the digitisation of the staff and product details, that ensures

* better management of restaurant resources,
* permits and oversees the user login by manager,
* simplicity in creating, retrieving, editing, updating and deleting the data in database
* quick, easy and systematic order input by touch on the UI,
* answering queries about product details to the customers,
* swiftness in calculation of order and transaction process,
* instant notification to the kitchen about orders,
* faster work flow, thereby
* better customer service with less queue time for order and delivery, thereby
* more sales implying more earnings,
* proper utilisation of workforce meaning more work is done by limited staff compared to addition of more staff needed to do the same amount of work in same timeframe while using traditional ordering method,
* instant recording of sales compared to need of extra time and personnel for the same purpose in traditional method.

1. **Development Methodology**

The development methodology refers to the Software Development Life Cycle (SDLC) of this system, which are described in detail as:

* 1. **. Methodology**

Modern Waterfall SDLC model, also known as Iterative Model, is utilized as a methodology for development of this software.

At first, a ‘Feasibility study’ was conducted by the developer team of the practicability of this software through discussions and research. After that followed ‘Requirement Analysis and Specification’ phase, which involved ‘Requirements Gathering and Analysis’, and ‘Requirement Specification’ creating a Software Requirement Specification (SRS) document, which documents the functional and non-functional requirements, aim, objectives, features, problem statements, scopes and details of the software. Subsequently, a structured ‘Design’ was finalized iteratively through object-oriented approach considering various objects in problem domain and solution domain along with their relationships. After that, the design is translated into source code using Python programming language in ‘ Development’ phase to build a working software, which is unit tested and integrated incrementally in the following ‘Integration and System-testing’. The software can now be released to users and regularly maintained in the last stage of ‘Deployment and Maintenance’. The above phases were revisited as per need due to new user requirements and feedbacks.

* 1. **. Tools and Technologies**

Several tools and technologies were used in the entire SDLC. They are listed as follows:

* **Notepad** and **MS Word** for ‘Feasibility Study’ and ‘Requirement Analysis and Specification’ phase,
* **Figma, Adobe Photoshop, Adobe Illustrator** and **GIMP** for ‘Design’ phase,
* **Python** with **tkinter library** and **Visual Studio Code** to write code in ‘Development’ phase
* **Sqlite3** for database in ‘Development’ phase
* **Git** and **Github** for version control
* **Google search engine** for research purpose.
  1. **Conceptual Diagram**

1. **System Architecture**
2. **Project Plan**
3. **Prototypes**
4. **Developed System**
5. **System Testing**
6. **Version Control**
7. **Conclusion**
8. **References**