**Fast Food Billing System**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the course project of

**CSE305L Software Engineering Lab**

Submitted by

Raavi Brahmendra(AP20110010629)

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[May, 2023]**

Table of Contents

[1](#_heading=h.gjdgxs) Introduction 4

[1.1](#_heading=h.30j0zll) Purpose 4

[1.2](#_heading=h.1fob9te) Scope 4

[1.3](#_heading=h.3znysh7) Objectives 4

[1.4](#_heading=h.2et92p0) Definitions, Acronyms, and Abbreviations 5

[2](#_heading=h.tyjcwt) System Overview 6

[2.1](#_heading=h.3dy6vkm) System Components 6

[2.1.1](#_heading=h.1t3h5sf) Menu Management 6

[2.1.2](#_heading=h.4d34og8) Order Management 6

[2.1.3](#_heading=h.2s8eyo1) Management Of Reports 6

[2.2](#_heading=h.17dp8vu) System Architecture 6

[2.3](#_heading=h.3rdcrjn) System Interfaces 6

[2.3.1](#_heading=h.26in1rg) Inventory Management 6

[3](#_heading=h.lnxbz9) Data Flow Diagrams 7

[3.1](#_heading=h.35nkun2) Context Diagram 7

[3.2](#_heading=h.1ksv4uv) Level 1 Diagrams: 7

[4](#_heading=h.44sinio) Entity-Relationship Diagrams 8

[5](#_heading=h.2jxsxqh) Functional Requirements 9

[5.1](#_heading=h.z337ya) Use Case Diagram 9

[5.2](#_heading=h.3j2qqm3) Use-Case Scenario 10

[6](#_heading=h.1y810tw) Activity Diagrams 11

[7](#_heading=h.4i7ojhp) Class Diagrams 12

[8](#_heading=h.2xcytpi) Sequence Diagrams 13

[9](#_heading=h.1ci93xb) User interface ( prototype) 14

[10](#_heading=h.3whwml4) Non-functional Requirement 15

[10.1](#_heading=h.2bn6wsx) Performance Requirements 15

[10.2](#_heading=h.qsh70q) Usability Requirements 15

[10.3](#_heading=h.3as4poj) Security Requirements 15

[10.4](#_heading=h.1pxezwc) Reliability Requirements 15

[10.5](#_heading=h.49x2ik5) Scalability Requirements 16

[10.6](#_heading=h.2p2csry) Maintainability Requirements 16

[11 Reports 20](#_heading=h.2grqrue)

[12](#_heading=h.vx1227) References 21

# Introduction

In this detailed document, it consists of specification and features of the software. It helps to understand the target audience and user classes accordingly and function and non-functional requirements.

## Purpose

The purpose of the document is to set forth the requirements of developing the fast-food billing system

The fast-food industry has expanded quickly in recent years all across the world, which makes the billing process difficult.

At peak time, the crowd would be twice as expected; pen and paper invoicing methods still used today result in uncomfortable wait times.

If the lunch breaks are restricted, the Employees problems can have problems, and it might be challenging to bill for the time. They typically have to skip lunch or perform an independent job in these situations.

Our opinion is that, in light of aforementioned issues and current technological advancements, a semi-automatic system is necessary. It makes it convenient for clients or the intended audience to eat in, get takeout and unwind.

## Scope

This documentation will be addressing each component of the fast-food billing system, along with the requirement specification, the layout of the user. interface, system architecture, and both functional and non-functional needs. Also, there are numerous other choices, which enables clients to order and choose their meals. pre food billing offers "budget, which is a significant Component for the aforementioned groups. This is since, while analysing our target audience, middle class, students, and workers, budget is an important factor to Consider. The system's implementation features Will not be addressed.

## Objectives

Before initiating development, this SRS document is meant for all stakeholders to review and confirm all the requirements and features. The sponsors, management team, product champions, and Q/A crew are Examples of stakeholders. It will be. Presented in an approachable manner with a thorough description of features and specs. The software also specifies the operational condition under which the system should function as well Some restrictions and limitations.

## Definitions, Acronyms, and Abbreviations

SRS – Software Requirements Specification

UI – User Interface

API – Application Programming Interface.

# System Overview

The fast-food billing system will function independently of all other systems as a stand-alone system. To process electronic payments, the system will connect with such an e-payment gateway. To force the user to complete the transaction using the scanner, the system will generate a QR code on the bill and provide the token number.

## System Components

### Menu Management

The system allows for the management of menu items, pricing and descriptions by restaurant personnel.

### Order Management

With the system, restaurant Personnel will be able to handle orders, including updating their status, monitoring their progress and cancelling them. Customers will be able to Place orders online.

### Management Of Reports

The system will be enabled restaurant personnel to produce sale, inventory, and client order reports on daily, weekly and monthly basis.

## System Architecture

The Fast-Food Billing System will be built using a three-tier architecture. The first tier will consist of the presentation layer, which will handle the user interface. The second tier will consist of the business layer, which will handle the logic of the application. The third tier will consist of the data layer, which will handle the data storage and retrieval.

## System Interfaces

### Inventory Management

With the system, Restaurant workers will be able to monitor Inventory levels, handles inventory orders and get notifications when levels are low.

The fast-food billing system will integrate with existing pos system payment gateways and inventory management software.

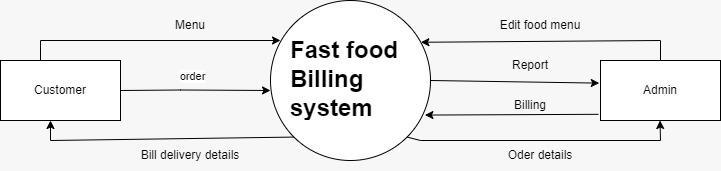
The system will also use APIs to communicate with these external systems.

My SQL database, JavaScript for front end and PHP for backend.

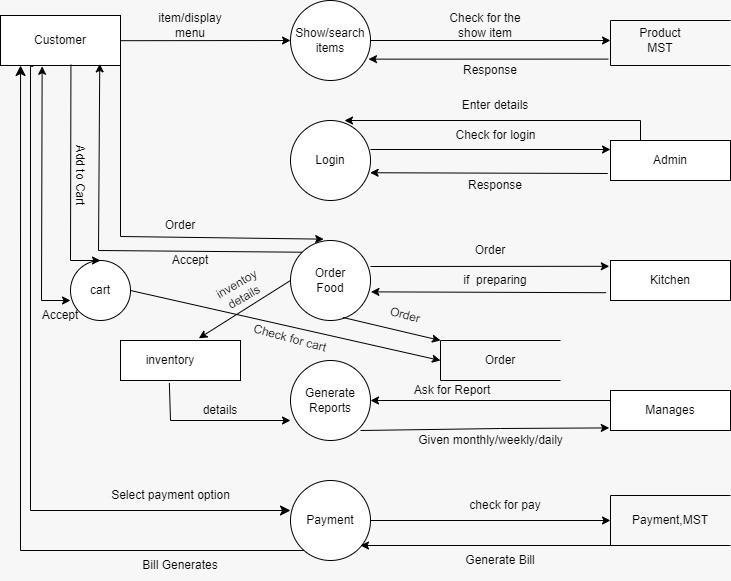
A cache would be required to store data.

# Data Flow Diagrams

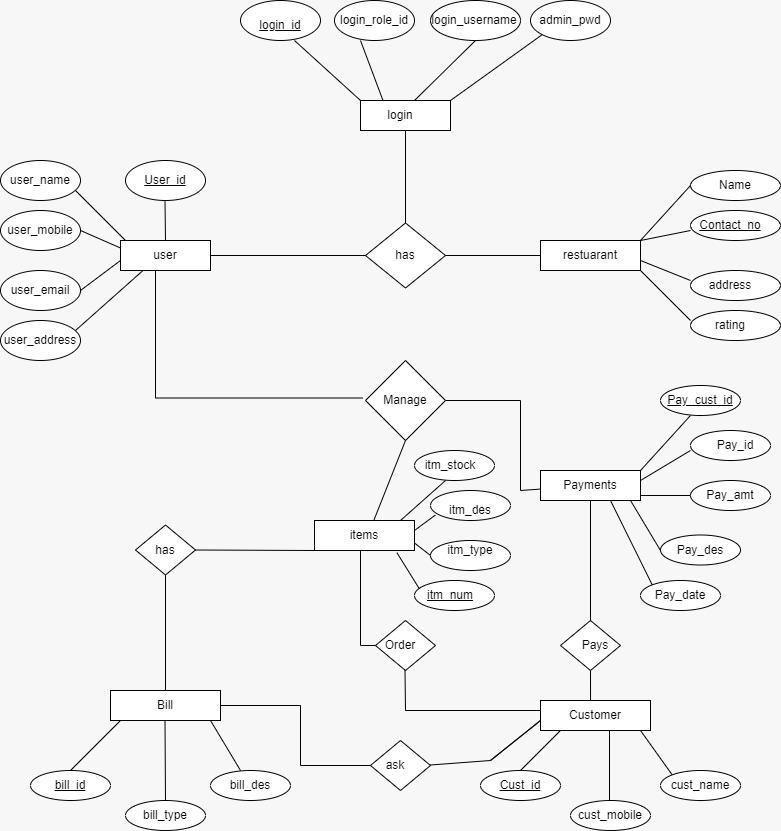
## Context Diagram



## Level 1 Diagrams:

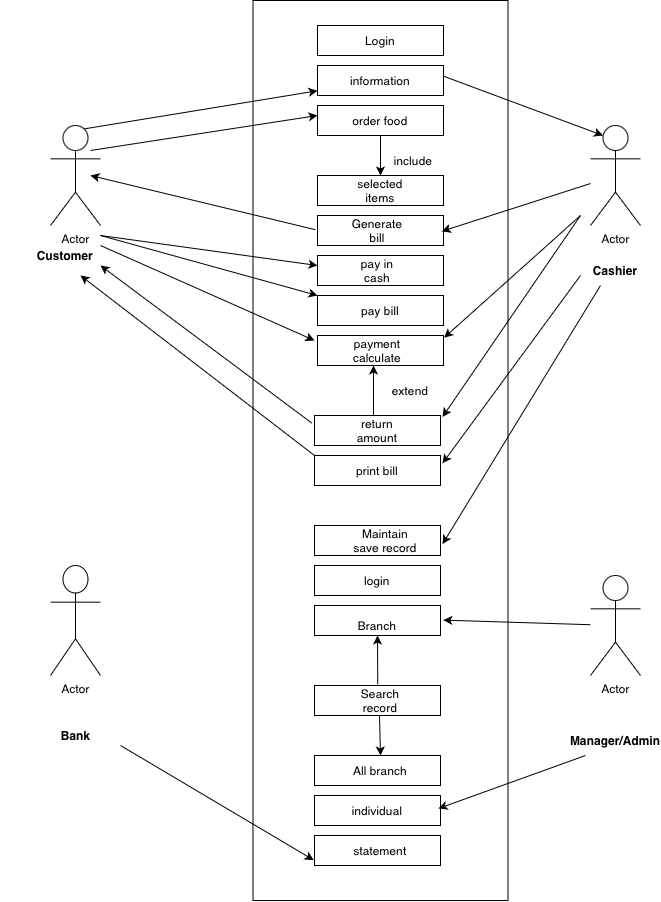


# Entity-Relationship Diagrams



# Functional Requirements

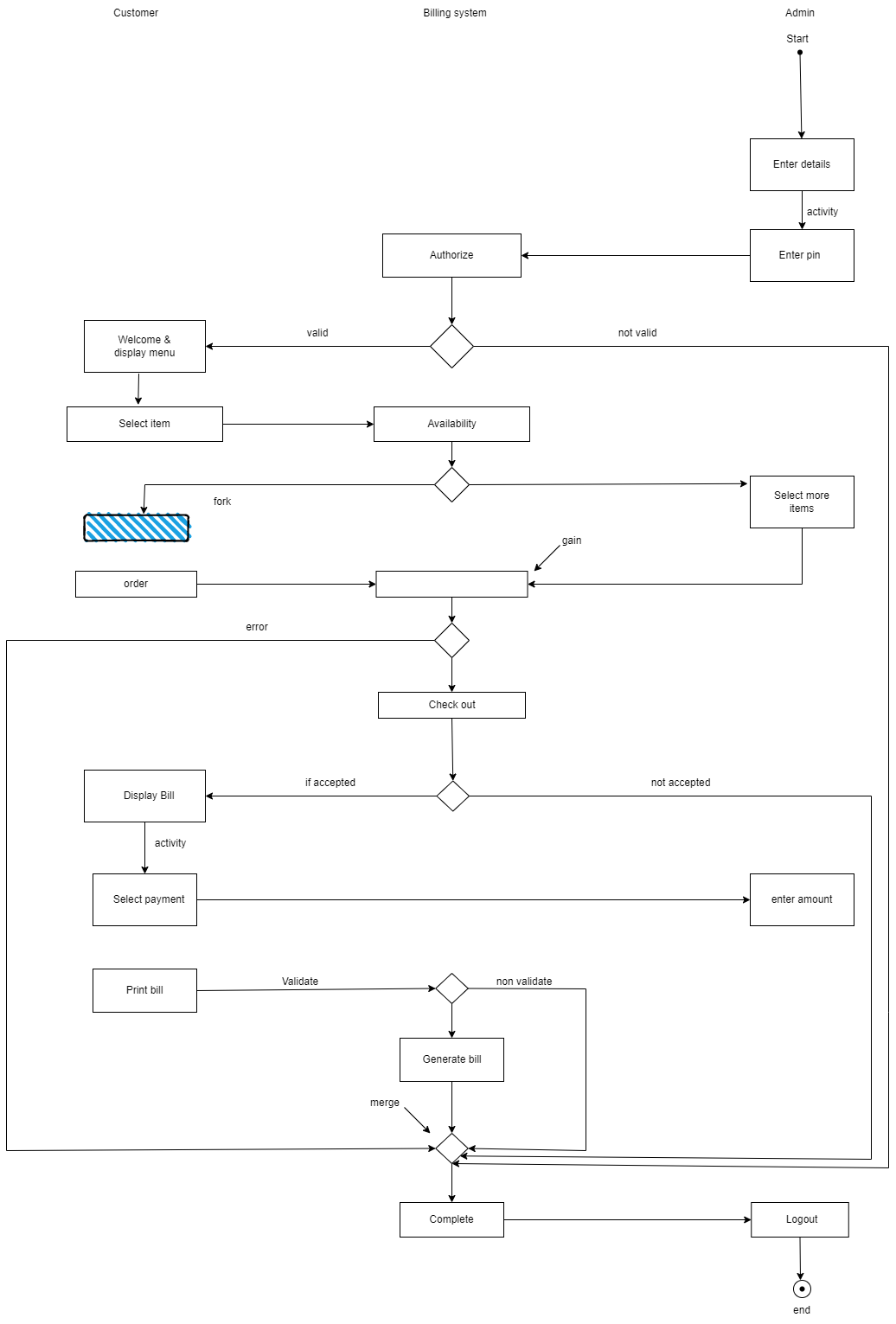
## Use Case Diagram



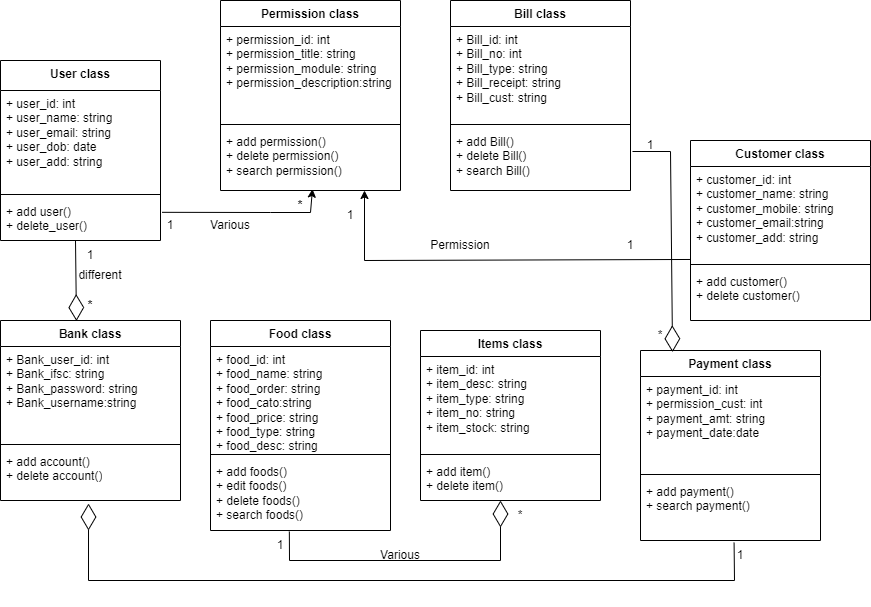
## Use-Case Scenario

* Admin gives access to cashiers to login.
* Then cashier enters the access code to access the system.
* System enables secure communications.
* System displays restaurant menu card to the customer.
* The Customer selects the items that are included in menu and order-food.
* Cashier generate bill and collect the amount.
* Amount Can either be collected through cash or online payment via phonepe, Paytm or card.
* If Customer pays through cash, cashier calculate the payment and return the remaining amount if required.
* Cashier prints bill and customer will collect bill.
* All the records are saved and maintain by admin or cashier.
* If admin wants to search particular amount, he can be done.
* After billing, transaction number in displayed by system.
* System displays message to visit again.

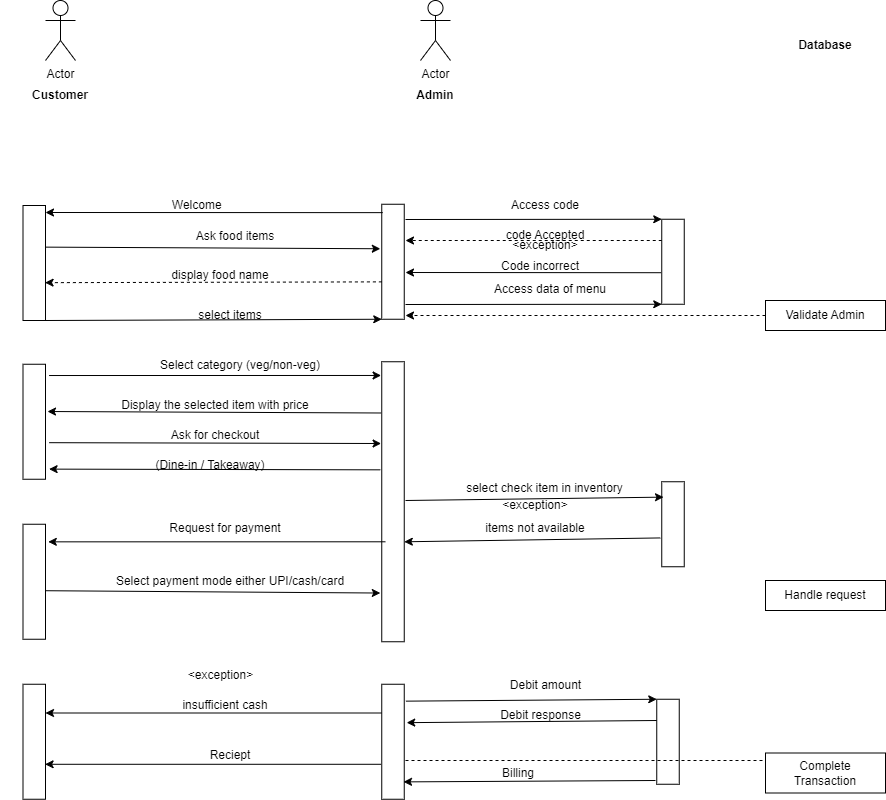
# Activity Diagrams



# Class Diagrams



# Sequence Diagrams



# User interface ( prototype)

The user interface of the fast-food billing system should be easy to navigate and simple to use for the customers and the staff. The following are the basic requirements for the user interface:

* The system should have a login screen for the staff to enter their credentials and access the system.
* The system should have a menu screen that displays the menu items along with their prices, descriptions, and images.
* The system should have an ordering screen that allows the customers to select the items they want to order and add them to the cart.
* The system should have a payment screen that displays the total amount due and the different payment options available to the customers.
* The system should have a receipt screen that displays the items ordered, the total amount paid, and any other relevant information.

# Non-functional Requirement

## Performance Requirements

* The system should be able to handle at least 1000 transactions per day without any significant slowdown.
* The system should generate bills and receipts within 2 seconds of receiving payment.
* The system should be able to generate reports on demand without significant delay.

## Usability Requirements

* The system should be user-friendly and easy to use for both staff and customers.
* The interface should be intuitive and easy to navigate.
* The system should be able to accommodate different languages, currencies, and payment methods.

## Security Requirements

* The system should have secure login credentials for staff members with different levels of access.
* The system should keep all customer and financial data secure and confidential.
* The system should be able to detect and prevent fraudulent activity.

## Reliability Requirements

* The system should be able to handle high traffic and not crash or freeze.
* The system should have a backup and recovery plan in case of power outages or other disruptions.
* The system should be able to process transactions accurately and reliably.

## Scalability Requirements

* The system should be scalable and able to handle increased traffic as the business grows.
* The system should be able to accommodate new products and menu items without significant changes to the system architecture.
* The system should be able to integrate with other systems as needed.

## Maintainability Requirements

* The system should be easy to maintain and update by IT staff.
* The system should have a clear and comprehensive user manual and technical documentation.
* The system should have regular updates and bug fixes to ensure smooth operation.

# Glossary

* POS: Point of Sale
* SKU: Stock Keeping Unit
* VAT: Value Added Tax
* UI: User Interface
* API: Application Programming Interface

# References

* Martin, R. C. (2014). Clean architecture: A craftsman's guide to software structure and design. Pearson Education.
* Sommerville, I. (2015). Software engineering. Pearson Education.
* UML Documentation. (n.d.). Retrieved from <https://www.uml-diagrams.org/index-examples.html>