SRM Institute of Science and Technology

18CSC206J

Software Engineering and Project Management



Restaurant Bill Management System

Problem Description

This project helps people, business including hotels to have a standard building system. It is a more reliable and faster method of billing and tractions through ‘QR’ and ‘Graphic User Interface (GUI)’.

History

Earlier Building systems were less efficient as some items were not registered as well as there was a manual way of paying through cash with this standard system it is easier to pay using the ‘QR’ code.

Limitation

It is only limited to hotel cafeteria this standard billing system

Approach

Collecting data of expenses through ‘GUI’ and ‘QR’ code.

Benefits

This system help customers transactions easier rather than

Making customers waste their time waiting for their bill as well as making the bill process more efficient and faster.

Since the start of QR the machine take time to generate the ‘QR’ code for transactions this system will generate the ‘QR’

Faster and transactions will take less time.

Since the start of ‘QR’ the machines take time to generate the ‘QR’ code for the faster and transactions will take less time.

Purpose and Need

The Billing system helps to manage items in a more organised manner, it is a more efficient system as transactions will be faster in restaurant through and bills will be generated automatically on your devices so the moment process of waiting for bill and wasting time will be cut-off.

Goals & Objectives

The objective is not the keep the customers waiting for the bill as well as to remove the change system through QR transaction.

Audience

Daily customers, Foodies, small business and hotels for adopting the billing System.

Existing Theories

Data of items in the menu

Standard prices with G.S.T of each item in the menu

Feasibility

The project is done “free of cost”.

Availability of Existing data

There is homogenous data, but we need to organise it and use it for giving better result.

What type of useable informational tools are available and practical

We will use Graphic User Interface (GUI) using python and a QR code generator

Problem Statement

Traditional methods of bill management in restaurant were ineffectual and effective there was a major scope of error for example waiters used to take the order of the customers on paper, which would to lead to misunderstandings between them would lead to conflicts related to total bill amount and at the delivery time or pickup time the right item to the right customer, After this the customer need to wait for a while to get the bill and there could be some miscalculations errors in the bill.

Now to overcome these problems we came up with a model known as ‘Restaurant Bill Management System’ which can solve most of the problems we faced in the regular method, this method is very efficient and there is minute scope of error. It can overcome calculations error as the system performs calculations very accurately. The customers need not have wait for the bill as they get it at the time of ordering food and customers can also calculate and order according to their budget. This method can also solve the conflicts we faced in the regular method.

Who is affected positivity and negatively by the project?

People who are trying to increase their business in hotels and restaurant without getting loss or any negative feedback from the

Customers due to delay I taking order and waiting a while for getting bill so people who own restaurants are affected positivity

The game restaurants will be affected negativity if there is a fault or error and they might get loss if the server of the applications is down to any reason.

Who has the power to make it successful or fault?

People who own a restaurant if they have proper knowledge of the software/application it becomes successful.

If the person do not have any proper knowledge then it might fail.

Who makes the decisions about money?

As we are making the software without using money, no money is involved.

Who are the supplies?

We who created the software are the supplies.

Who are the end users?

Restaurant owners are the ones who’s going use this software. Cashiers in the Restaurant use this software mainly.

Who has influence over the software?

Faculties and other staff members.

Who could solve potential problems with the project?

Potential Problems with this are:-

* Not uploading the software frequently
* Entering wrong data

Who’s in charge of assigning or procuring resources or facilities?

Stakeholders of the project.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stake Holder Register** | | | | |
| Project Name- Restaurant Bill Management | | | | |
| **Prepared By- Ayush Gupta, Sai Darshan, Jaiaditya** | | | | |
| ProjectStakeholder | Specific information Needs | ProjectInterest | Impact on Project | Role |
| **Name** | **Types and frequency of communication** | **Specifications of interests and participation** | **Positive,****Negative****Influence** | **Decision Making,****Collaboration** |
| Dr. Shiny Angle T.S | Live Meeting and Video Conference | Control, Tracking Management of Project | Influencer | Project Manager and Review |
| **Ayush Kr. Gupta** | **Live Meeting** | **Testing and Integration** | **Positive** | **Testing and Developer** |
| Sai Darshan Reddy | Live Meeting | Python(GUI)Library | Positive | Fronted Developer |
| **JaiAditya****Ghorpade** | **Live Meeting** | **Logical and Algorithm design** | **Positive** | **Algorithm Developer** |

Project Model

What Programming the project requires and current limitations are reason for choosing V-Model project management methodology

Below are some reasons that illustrate the chosen model

* Approach - Sequential Manner
* Success Management - Business Value
* Project Size - Small
* Management Style - Decentralized
* Perspective to change - Change adaptability
* Culture - Leadership-Collaboration
* Documentation - Low
* Emphasis - Business Oriented
* Cycles - Numerous
* Domain - Exploratory
* Upfront Planning - Medium
* Return on investment - Freely in Project
* Team Size - Small

|  |  |
| --- | --- |
| Experiment No:- | 3 |
| Title of Experiment | System, Functional and Non-Functional Requirements of the Project |
| Team Members | Ayush Kumar Gupta  K.Sai Darshan Reddy  Jai Aditya Ghorparai |
| Register Number | RA2011026010038  RA2011026010036  RA2011026010035 |
| Date of Experiment | **12-04-22** |

Mark Split Up

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Description | Maximum Mark | Mark Obtained |
| 1 | Exercise | 5 |  |
| 2 | Viva | 5 |  |
| Total | | 10 |  |

**Staff Signature with date**

# Aim

To identify the system, functional and non-functional requirements for the project.

**System Requirements**

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management ,allowing the company to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the company to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.

**Functional Requirements**

As can be seen in the system model diagrammed above, each of the three system components essentially provides a layer of isolation between the end user and the database. The motivation behind this isolation is twofold. Firstly, allowing the end user to interact with the system through a rich interface provide a much more enjoyable user experience, particularly for the nontechnical users which will account for the majority of the system’s users.

* Waiter/Server
* Customer
* Supervisor
* Accountant/ Cashier

**Non-Functional Requirements**

### Any Requirement That Specifies How the System Performs a Certain Function

### Performance – for example Response Time, Throughput, Utilization, Static Volumetric Scalability

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs

* Security
* Performance requirements
* Knowledge of software
* Reliability
* Usability
* Serviceability
* User Friendly

|  |  |
| --- | --- |
| Functional Requirements | Non-Functional Requirements |
| Product features helps hotels andRestaurant fun efficient and quick billingIt makes the process of taking orders and generating bills easierAccurate bill calculationsSystem shall display all food items offend by the restaurantSystem shall organise items categorically | **Product property agile systems of a penations and management** **System shall support till 5000 transactions per day****Systems shall not lose any transactions data****Systems will be available 24/7 and for 365 days****System shall accept payment and noise an order with seconds in 95 % of the cases** |

# **SESSION 4**

Project Plan, Effort & Roles-Responsibilities

# Aim

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

|  |  |
| --- | --- |
| Focus Area | Details |
| Scope Management | **A GUI based restaurant billing system using in software management****Requirement of python and its various inbuilt modules and libraries****Schedule to small scale hotels, restaurants as well as big food chains****Activities Development of model with minimal error** |
| Schedule Management | **Problem Description****Problem Statement****Project Model****System Requirements** |
| Cost management | **₹2, 08,500** |
| Stakeholder | **User and Professor are the Stakeholders** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effort and Cost Estimation** | | | | |
| AvidityDescription | Sub-Task | Sub-TaskDescription | Effort (Hours) | Cost (INR) |
| **Design the user screen** | **E1R1A1T1****E1R1A2T2** **E1R1A1T3** | **Confirm the user requirements****Creating the interface****Dry run testing** | **5** **4** **6** | **₹ 1000****₹ 2000****₹ 1500** |
| Identify data source for displaying unit of energy Consumption |  | Training and Technical supportImplementation and Operation | 66 | ₹ 2000₹ 2000 |
|  |  |  | **26** | **₹ 8500** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Infrastructure / Resource Cost** | | | |
| InfrastructureRequirement | Quantity | Cost PerQuantity | Cost per (Item) |
| **Computer****Mobile / Tablet** | **2****4** | **₹ 50,000****₹ 25,000** | **₹ 1, 00,000****₹ 1, 00,000** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Team Formation** | | | | |
| Name | Role | Responsibility | | |
| **Dr. Shiny Angel T.S****Sai Darshan Reddy****Ayush Kr Gupta****Sai Darshan Reddy****Sai Darshan Reddy****Ayush Kr. Gupta** | **Key Business User Project Manager****UX Designer** **Front-End Developer**  **Back End Developer** | **Provide Clear Business requirements** **Manage the project** **Creating the interface****Develop User Interface****Design, Develop and Unit Test Services****API/DB** | | |
| Identify data source for displaying unit of energy Consumption |  | Training and Technical supportImplementation and Operation | 44 | 20002000 |
|  |  |  | **17** | **8500** |

# **SESSION 5**

Work breakdown structure

Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI No** | **Register No** | **Name** | **Role** |
| **1** | **RA2011026010038** | **Ayush Kumar Gupta** | **Rep** |
| **2** | **RA2011026010036** | **K.Sai Darshan Reddy** | **Member** |
| **3** | **RA2011026010035** | **Jaiaditya Ghorpade** | **Member** |

**Work Breakdown Structure**

* 0.0 Billing system
* 1.0 Project Management
* 2.0 Requirements Gathering
* 2.1 Hardware requirements
* 3.0 Analysis & Design

3.1 Software Requirements

* + - 3.1.1 Database Implementation
    - 3.1.2 Middleware Development
    - 3.1.3 Security Subsystems
    - 3.1.4 Catalog Engine
    - 3.1.5 Transaction Processing
  + 3.3 Graphics and Interface

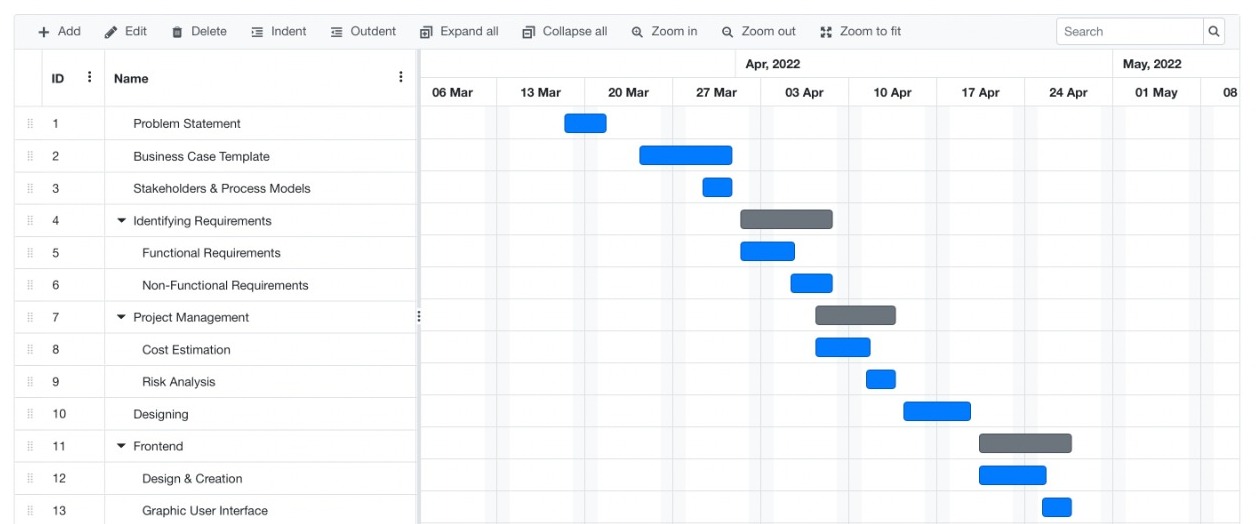
3.3.1 GUI (using python)

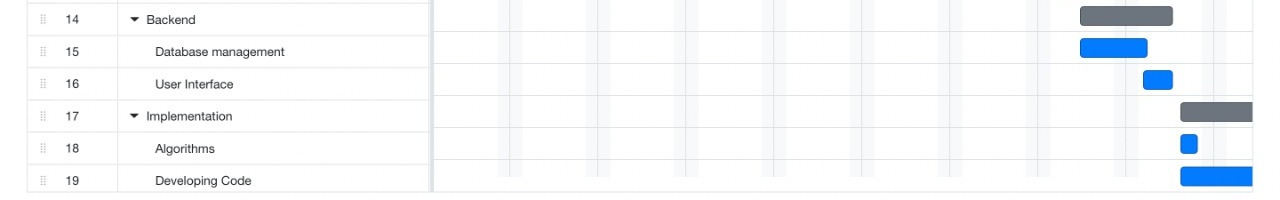
* 5.0 Testing and Production

**Flow Chart**

Swot Analysis

**Gantt chart**





|  |  |  |  |
| --- | --- | --- | --- |
|  | **RMMM MODEL** | | |
| **RISKS** | **RESPONSE** | **STRATEGY** | **EXAMPLES** |
| **Insufficient Memory** **Logical Warning** | **Avoid** | **Risk avoidance is a strategy where the project team take****Action to Remove the threat of the risk or protect from the impact** | **Remove unnecessary data****Increasing Test****Dynamic IP Address** |
| **No Backups of Files** | **Transfer** | Risk transference involves shifting or transferring the risk threat and impact to a third party. Rather transfer the responsibility and ownership | Purchasing insuranceWarranties and Contracts issuing |
| **Time complexity Error****Complex Model** | **Mitigate** | **Risk mitigation is a strategy were by the project team takes an action to reduce the probability of the risk occurring** | **Increasing Testing** **Reduce Process Complexity** **Easy and Feasible model** |
| **Insufficient Planning of Budget** | **Accept** | **Risk acceptance means the team acknowledges the risk and its potential impact but decides not to take any pre-emptive action to prevent it .It is dealt with only if it occurs** | **Reserve Budget****Schedule Management** |

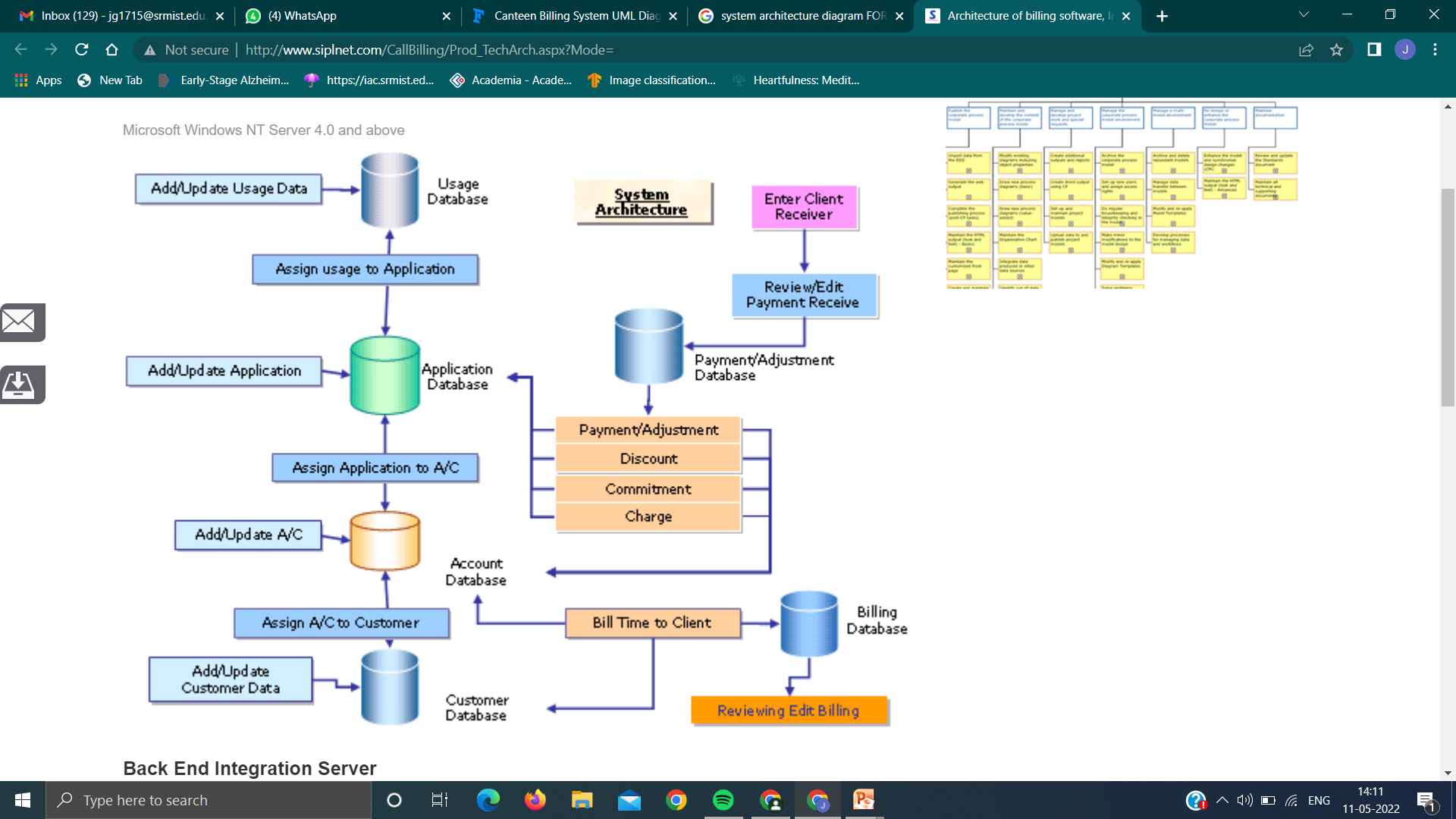
**Result:**

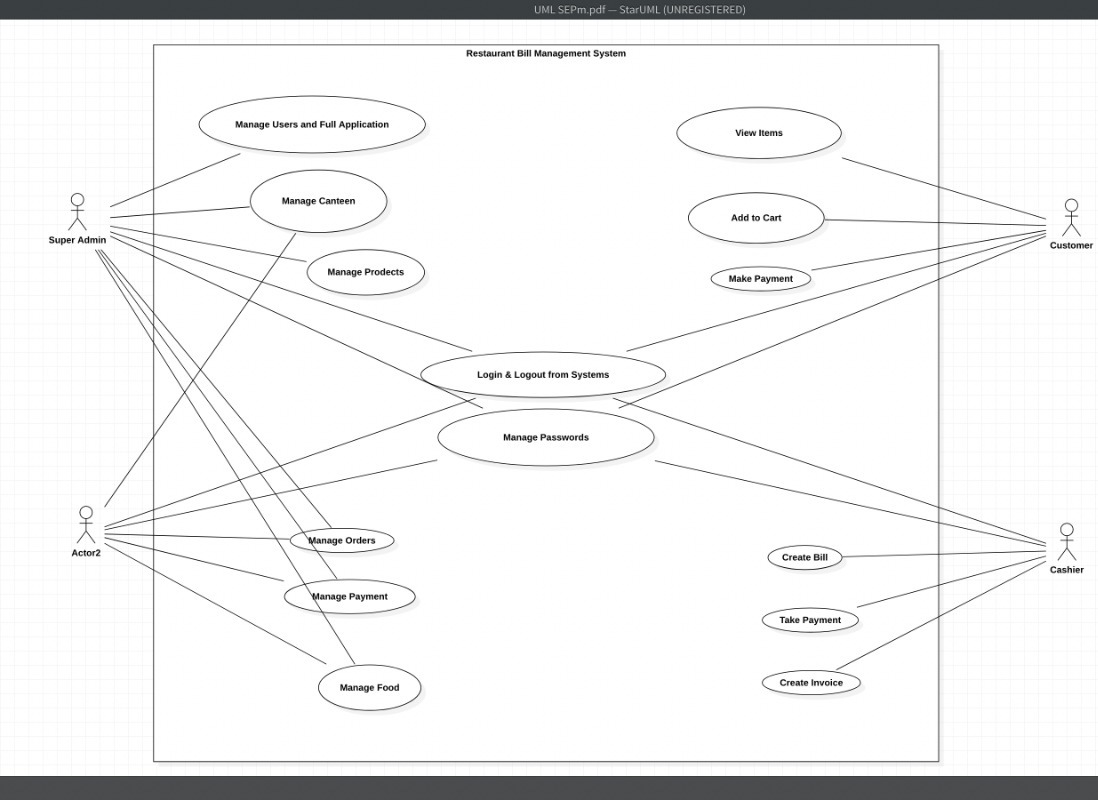
Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.

# **SESSION 6**

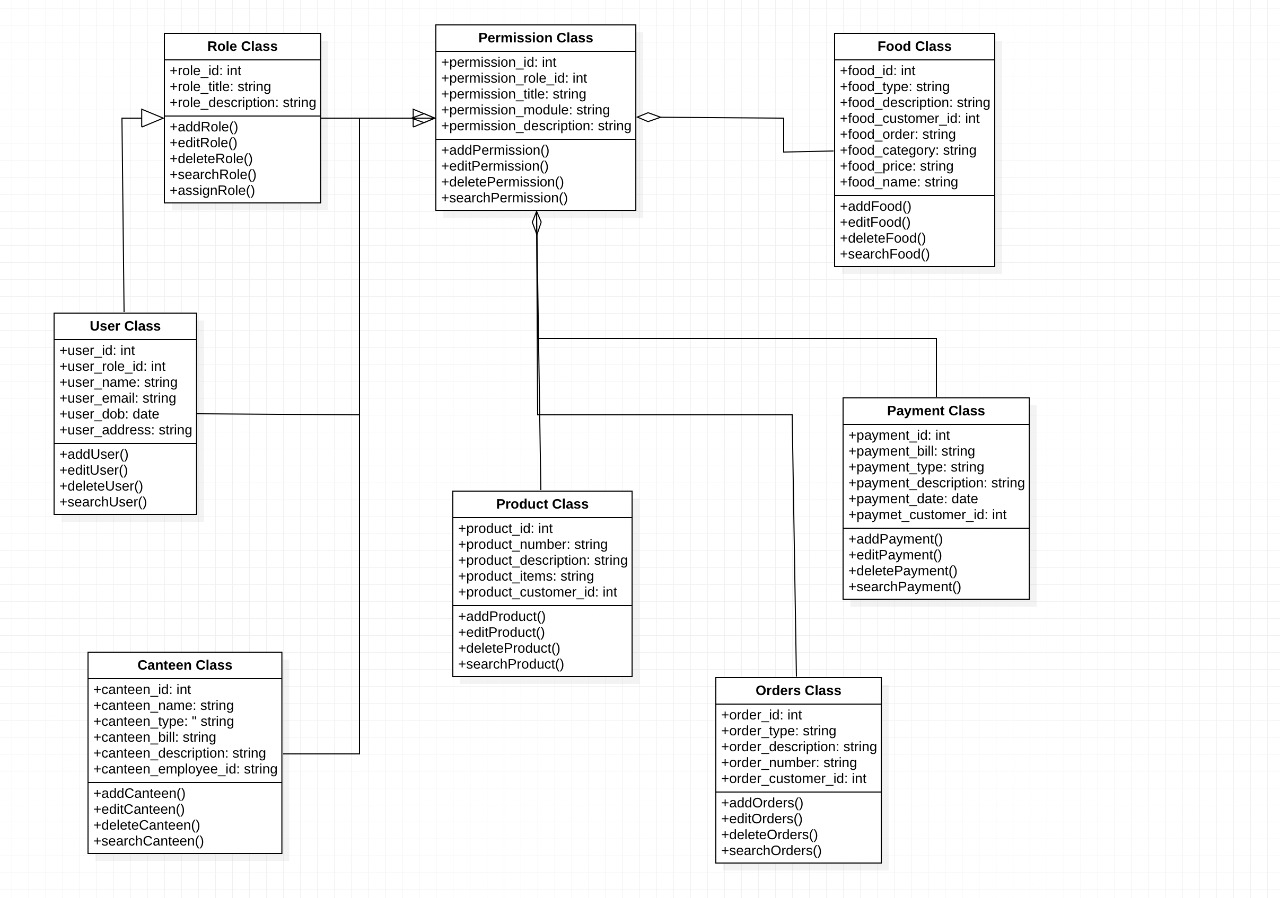
Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table



**Use-Case Diagram**

Class – Diagram



# **SESSION 7**

Aim

Design an Entity relationship diagram

**What is ER Diagram?**

* ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
* ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
* At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

**What is ER Model?**

* ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyse data requirements to produce a well-designed database.
* ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
* ER Modelling helps you to analyse data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modelling before implementing your database.

**Why use ER Diagrams?**

* Here, are prime reasons for using the ER Diagram
* - Helps you to define terms related to entity relationship modelling
* - Provide a preview of how all your tables should connect, what fields are going to be on each table
* - Helps to describe entities, attributes, relationships
* - ER diagrams are translatable into relational tables which allows you to build databases quickly
* - ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
* - The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram

- ERD Diagram allows you to communicate with the logical structure of the database to users

**Components of the ER Diagram**

This model is based on three basic concepts: Entities, Attributes, and Relationships

**ER Diagram – Notations**

- Rectangles represent entity sets.

- Diamonds represent relationship sets.

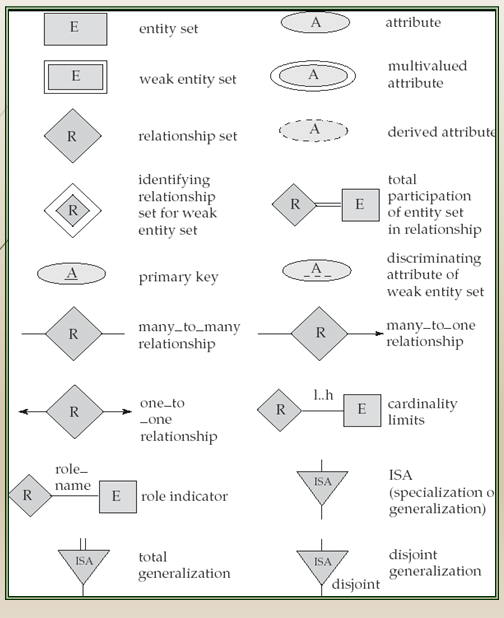
- Lines link attributes to entity sets and entity sets to relationship sets.

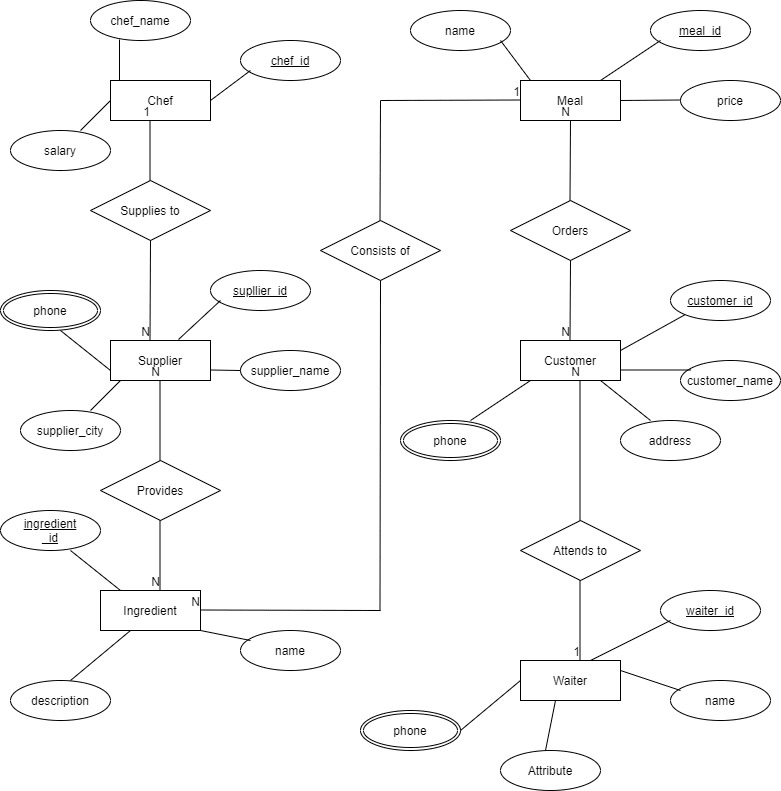
- Ellipses represent attributes

- Double ellipses represent multivalued attributes.

- Dashed ellipses denote derived attributes.

- Underline indicates primary key attributes



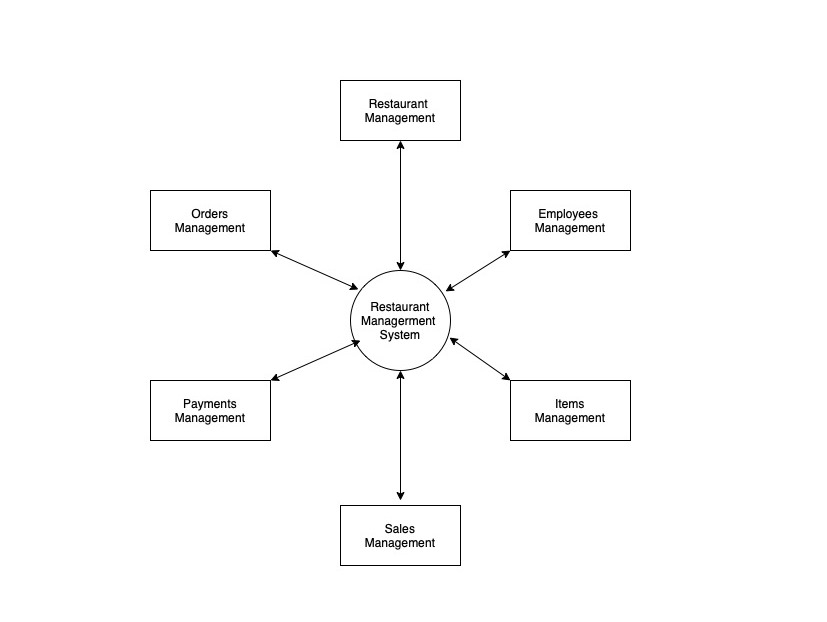


# **SESSION 8**

Aim

Develop a Data Flow Diagram (Process-Up to Level 1)

Level 0



**Data Flow Diagram**

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:

(1) Level 0 data flow diagram should depict the software/system as a

Single bubble;

(2) Primary input and output should be carefully noted;

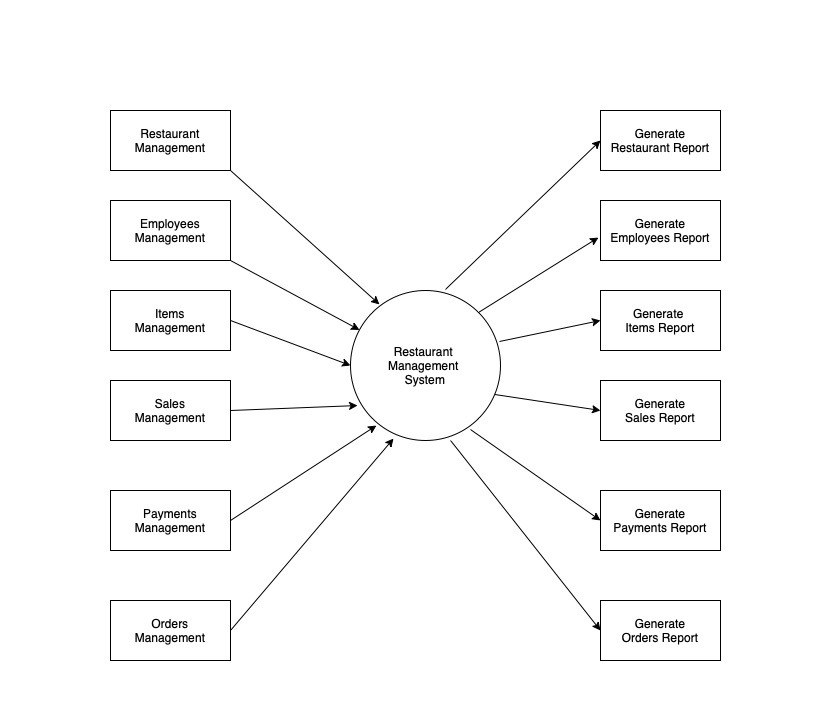
(3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;

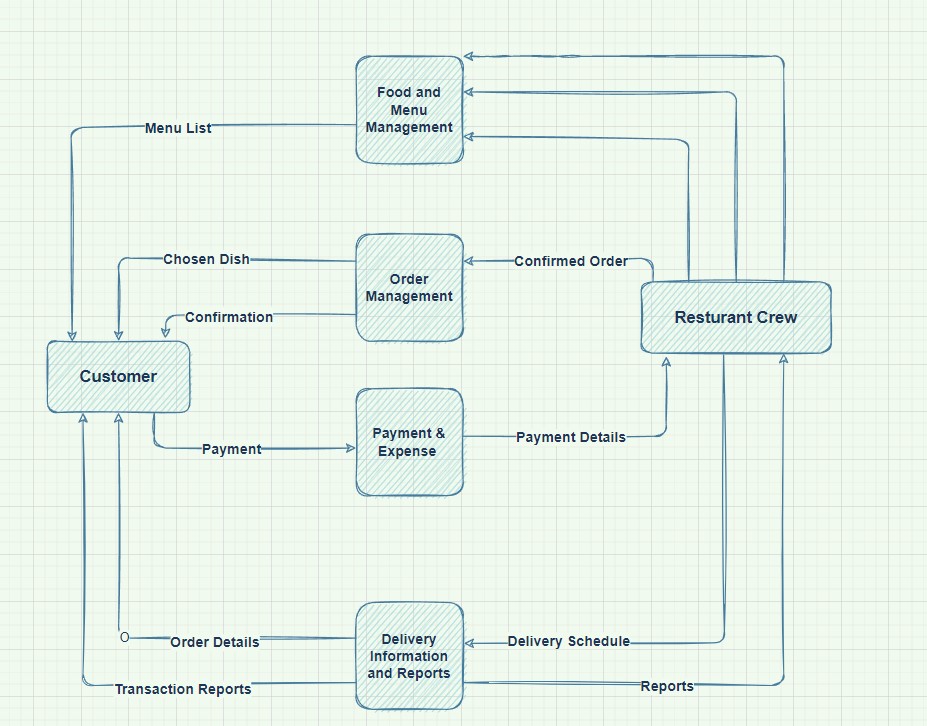
(4) All arrows and bubbles should be labeled with meaningful names;

(5) Information flow continuity must be maintained from level to level and

(6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

0

Level 1

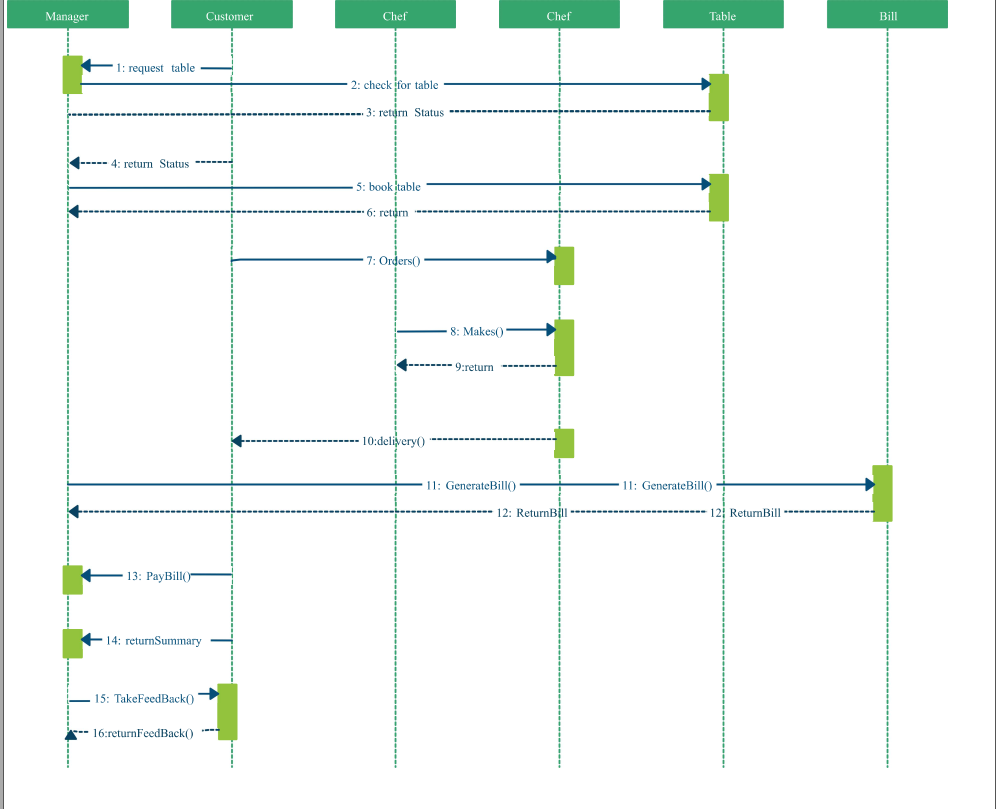
Level 1

# **SESSION 9**

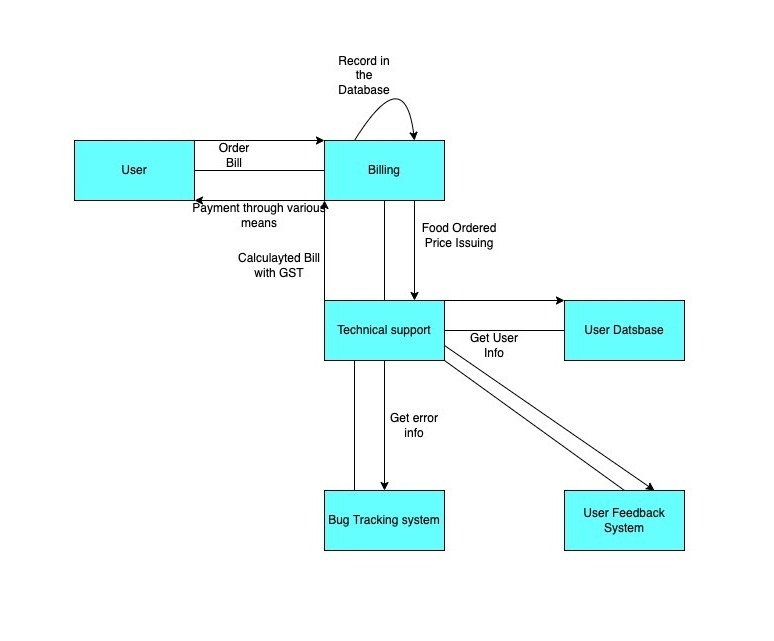
Aim

To Develop in a sequence and collaborate diagram

**Sequence Diagram**



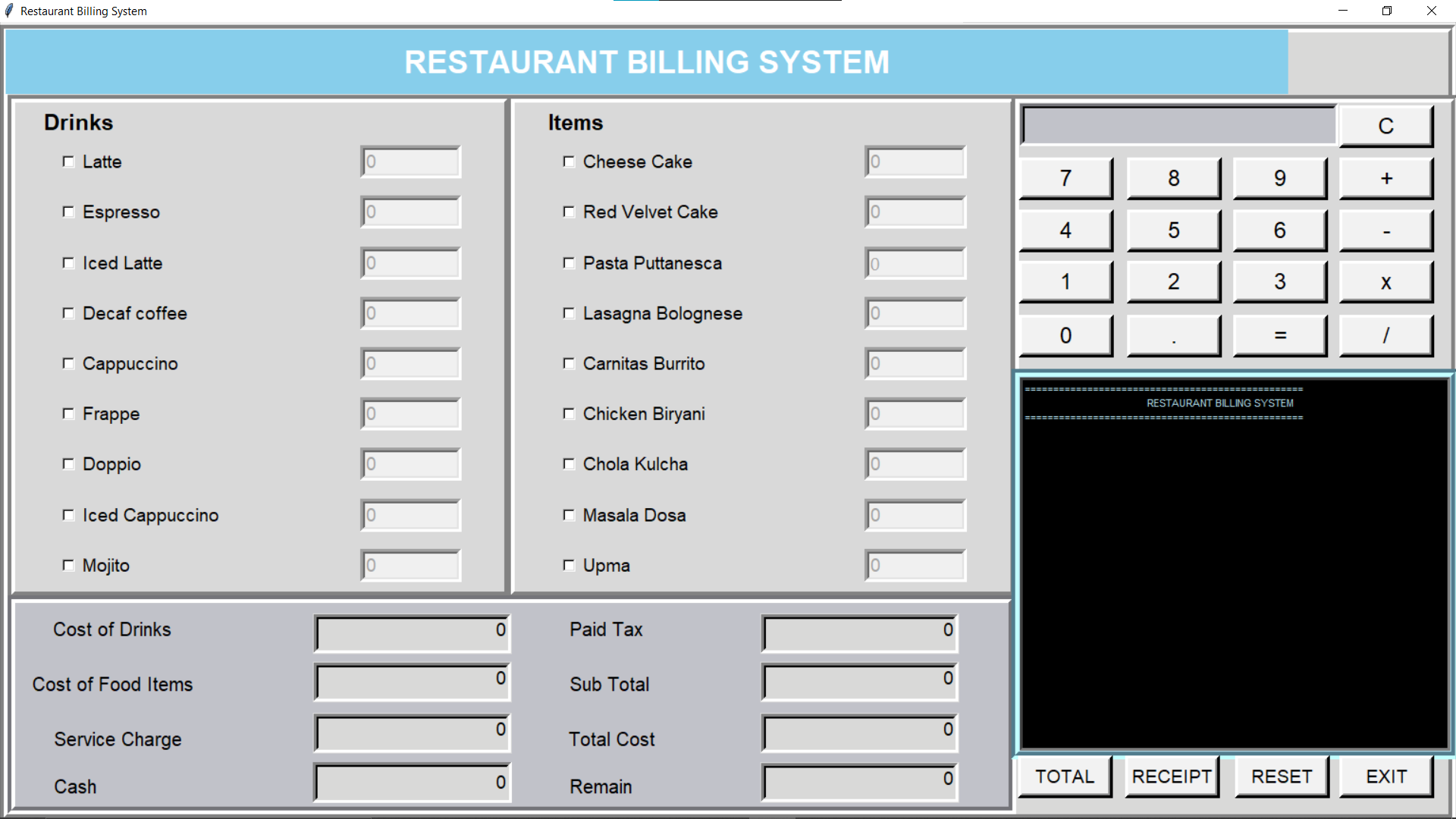
**Collaboration Diagram**



# **SESSION 10**

Aim

Develop a Testing Framework/User Interface



Result

Thus, the testing framework/user interface framework has been created for the Restaurant Bill Management System.

# **SESSION 11**

# **Test Cases**

Aim

To develop the test cases manual for the Restaurant Bill Management System

**Functional Test-Cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Execution Steps** | **Expected Outcome** | **Actual Outcome** | **Status** | **Remarks** |
| 1.  2.  3. | Waiter/server taking the order  Supervisor check for calculations  Accountant/ Cashier checks for the receipt | Accept inputs from a user  Calculation  Generating Bill/Receipt | 1. The user clicks on the check button for ordering 2. Enter the amount of the product 3. The user gives input to the   Software.  1. The user checks  arithmetic  calculations  2. Check run 2  Dry runs of  Calculation  1. The bill is generated  2. The bill is printed out | The user should be taken to the next page for entering more user details  After ordering the calculations are made  The bill is generated and printed | The User is taken to the next page.  Calculations are made and the total amount is displayed  The bill is generated and printed out | Pass  Pass  Pass | Success  Success  Success |

**Non-Functional Requirements:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Execution Steps** | **Expected Outcome** | **Actual Outcome** | **Status** | **Remarks** |
| 1.  2.  3.  4. | Security  Maintainance  Usability  Availability | Transaction security  Maintainanceof data and the application  Easy formatting of the code  Availability in restaurants and small hotels | User's bank and transaction details must be kept safe.  The customer’s bill should be calculated fastly  Commit code,  Run action,  Pull formatted code  List the items and their quantity,  Calculate the amount,  Print the bill. | This info is kept safe from unknowns.  The details of the transaction is to be saved for further calculation  User should be able to get the formatted code with ease  The customers will receive the bill just after ordering the food. | The information is kept securely.  The transaction details are saved for future reference.  Users will be able to get the formatted code with ease  The customers will receive the bill just after ordering the food. | Pass  Pass  Pass  Pass | Success  Success  Success  Success |

**Result:**

Thus, the test case manual has been created for the Restaurant Bill Management System

# **SESSION 1****2**

# **Manual Test Case Reporting**

Aim

To prepare the manual test case report for the Restaurant Bill Management System

**Functional Test-Cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Execution Steps** | **Expected Outcome** | **Actual Outcome** | **Status** | **Remarks** |
| 1.  2.  3. | Waiter/server taking the order  Supervisor check for calculations  Accountant/ Cashier checks for the receipt | Accept inputs from a user  Calculation        Generating Bill/Receipt | 1. The user clicks on the check button for ordering 2. Enter the amount of the product 3. The user gives input to the   Software.  1. The user checks  arithmetic  calculations  2. Check run 2  Dry runs of  Calculation  1. The bill is generated  2. The bill is printed out | The user should be taken to the next page for entering more user details  After ordering the calculations are made  The bill is generated and printed | The User is taken to the next page.  Calculations are made and the total amount is displayed  The bill is generated and printed out | Pass  Pass  Pass | Success  Success  Success |

**Non-Functional Requirements:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Execution Steps** | **Expected Outcome** | **Actual Outcome** | **Status** | **Remarks** |
| 1.  2.  3.  4. | Security  Maintainance  Usability  Availability | Transaction security  Maintainanceof data and the application          Easy formatting of the code      Availability in restaurants and small hotels | User's bank and transaction details must be kept safe.  The customer’s bill should be calculated fastly  Commit code,  Run action,  Pull formatted code  List the items and their quantity,  Calculate the amount,  Print the bill. | This info is kept safe from unknowns.  The details of the transaction is to be saved for further calculation  User should be able to get the formatted code with ease  The customers will receive the bill just after ordering the food. | The information is kept securely.  The transaction details are saved for future reference.  Users will be able to get the formatted code with ease  The customers will receive the bill just after ordering the food. | Pass  Pass  Pass  Pass | Success  Success  Success  Success |

**Result:**

Thus, the test case report has been created for the Restaurant Bill Management System