**SSN College of Engineering**

**Department of Information Technology**

**UIT2211 – Software Development Project - 1**

2023–2024

**Requirement Engineering & Risk Management**

# TABLE RESERVATION SYSTEM SOFTWARE

**Objective of the project:**

* To develop a software that has a user interface with both login systems for the customer and the hotel manager.
* The customers should be able to see the number of tables and seats per table that are available to them.
* Automatic reservation of the table based on the number of seats should be done by the system.
* A notification confirming the booking of the table should be sent to the customer and the hotel manager when the booking is done.
* The status of the table should change from occupied to available when the customer pays the bill.
* Periodic refreshment of the system to display new seats must happen to display new availability.

**Requirements Specification (RS):**

A hotel wants to develop a table reservation system for their clients to avoid unnecessary waiting.

* The software should display number of tables and seats per table to the clients.
* The system should be able to automatically reserve a table based on the number of seats required.
* A booking notification should be sent to the client and the hotel manager.
* When a customer pays off his bill, the status of that table should change from occupied to available.
* The system should periodically refresh and display the new availability.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint**  **#** | **Epic** | **User Story**  **#** | **Requirement**  **/ User Story** | **Essential or Desirable** | **Description of the Requirement** | **Remarks** |
| 1 | Updating Data Structure | Queue data structures are used. | New customer/ Customer who has paid the bill. | Essential | Updates the queue with the customer data. | Working perfectly |
| 2 | Booking the appropriate seat | By adding data to a dictionary with seat number as key and customer data as value | Customer data and table number. | Essential | Updates the table with the data of customer. | Working perfectly |
| 3 | Billing | Confirmation from the user for the manager side and then pay the bill. | Table number, bill amount and customer data. | Essential | Generate bill for the particular table. | Working perfectly |
| 4 | Updating after billing | Customer data is removed from the queue and dictionary and seats are updated. | Customer paying bill | Essential | Updates the data structure after billing (Removal of data from queue and dictionary) and updating seat. | Working perfectly |
| 5 | Display of data in Manager side | Through login as manager option. | Information | Essential. | Customer data is displayed in a table format in the manager side. | Working Perfectly |

**Risk Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Data corruption or inconsistency in CSV files | High | Large | Implement robust error-checking and validation mechanisms when reading and writing to CSV files. Regularly backup CSV files. |
| 2 | Performance issues with large CSV files and complex data structures | High | Large | Optimize data structures (e.g., using queues) and implement efficient algorithms. Split large CSV files into manageable chunks if necessary. |
| 3 | Incompatibility issues with Python libraries | Medium | Large | Use well-supported and stable libraries. Test all libraries for compatibility with the latest Python version. |
| 4 | Data loss due to improper handling or file corruption | Medium | Medium | Implement regular backups and version control for CSV files. Ensure robust error handling and recovery mechanisms. |
| 5 | Project delays due to the complexity of data structures and UI development | Medium | Small | Break down the project into smaller, manageable tasks with realistic timelines. Use agile methodologies to adapt and refine the project plan as needed. |
| 6 | Unauthorized access to sensitive data in CSV files | Medium | Medium | Encrypt sensitive data within CSV files. Implement access controls and secure the application using authentication mechanisms. |
| 7 | Data breaches or cyberattacks | Low | Small | Regularly update the Python environment and libraries to patch vulnerabilities. Use firewalls, anti-virus software, and secure coding practices. |
| 8 | Resistance to change from the pharmacy staff. | Low | Small | Involve staff in the development process to gather input and ensure the system meets their needs. Provide training and support during and after implementation. |
| 9 | Lack of user engagement leading to underutilization of the system | Medium | Medium | Highlight the benefits of the new system and provide ongoing support to encourage its use. |
| 10 | Inefficiency in data retrieval and manipulation due to the use of complex data structures. | Medium | Medium | Optimize data access and manipulation functions. Provide training for developers on efficient use of queues. |