Restaurant reservation system

Requirements Engineering

RestaurantSYS

Submitted By: T00202430

Computing with Software Development

Date Submitted: dd/mm/yyyy

**Table of Contents**

[1. Introduction/overview 3](#_Toc474162299)

[2. Functional Components 4](#_Toc474162300)

[3. User Requirements 5](#_Toc474162301)

[4. System Requirements 6](#_Toc474162302)

[4.1. System Level Use Case Diagram 6](#_Toc474162303)

[4.2. Manage tables 6](#_Toc474162304)

[4.2.1. Add table 6](#_Toc474162305)

[4.2.2. Remove table 6](#_Toc474162306)

[4.3. Manage reservations 6](#_Toc474162308)

[4.3.1. Book reservation 6](#_Toc474162309)

[4.3.2. Cancel reservation 6](#_Toc474162310)

[4.3.3. Check-in reservation 6](#_Toc474162310)

[4.3.4. Regulate tab 6](#_Toc474162310)

[4.4. Manage reporting 6](#_Toc474162311)

[4.4.1. List reservations 6](#_Toc474162312)

[4.4.2. Analyse volume 6](#_Toc474162313)

[4.4.2. Analyse income 6](#_Toc474162313)

[5. System Model 7](#_Toc474162314)

[5.1. Level-0 DFD 7](#_Toc474162315)

[5.2. Level-1 DFD 7](#_Toc474162316)

[5.3. Level-2 DFD (Process P1: Title) 7](#_Toc474162317)

[5.4. Level-2 DFD (Process P2: Title) 7](#_Toc474162318)

[5.5. Level-2 DFD (Process P3: Title) 7](#_Toc474162319)

[6. Data Model (Class Diagram) 8](#_Toc474162320)

[6.1. Class Diagram 8](#_Toc474162321)

[6.2. Relational Schema 8](#_Toc474162322)

[6.3. Database Schema 8](#_Toc474162323)

[7. Conclusion 9](#_Toc474162324)

[8. Appendices 10](#_Toc474162325)

[8.1. Appendix A – Title 10](#_Toc474162326)

[8.2. Appendix B – Title 10](#_Toc474162327)

# Introduction/overview

**Normal** text here

# Functional Components

# User Requirements

# RestaurantSYS will perform table administration

## RestaurantSYS will allow to register a new table

## RestaurantSYS will allow to remove a table

# RestaurantSYS will manage reservations

## RestaurantSYS will allow a reservation to be made

## RestaurantSYS will allow a reservation to be cancelled

## RestaurantSYS will allow to check-in guests arriving

## RestaurantSYS will allow to regulate a payment

# RestaurantSYS will perform administrative reporting

## RestaurantSYS will allow to provide a list of daily reservations

## RestaurantSYS will provide a revenue analysis report

## RestaurantSYS will provide a table analysis report

# System Requirements

System will require to perform adding and removing of tables to database. Also must be able to manage reservations from booking to payment. Finally it should be capable of printing reports of income, business volume and list daily reservations.

## System Level Use Case Diagram

High level modules of system are represented here.

RestaurantSYS

Manage Tables

Manage Reservations

Guest

Manager

Manage Admin

## Manage Tables

This module will allow to manage tables to database.

It will be composed of function that will add tables and function that will remove tables.

### Add Table

This function will require user interaction to add tables to existing database.

<<includes>>

Manager

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Table** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will allow to add tables to a database | |
| **Preconditions** | Table must not exist in the database already | |
| **Trigger** | User action | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager invokes add table function  **Step 3:**Manager enters table details   * Table ID * Seats amount * Table description * Table status   **Step 4:**Manager confirms that the table is to be set | **Step 2:**System will display UI  **Step 5:**System validates the data entered   * All the data must be entered * Seats amount must be numeric * Table ID must be a number and greater than zero * Table status must be set * Table must not exist in the database already   **Step 6:** Table is added to the tables file |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager invokes add table function  **Step 3:**Manager enters table details | **Step 2:**System will display UI  **Step 4:**Invalid data is entered  **Step 5:**System displays error message |
| **Conclusions** | The new table is added to a database | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Remove Table

This function will require user interaction to remove tables from the database

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove table** | |
| **Use Case Id** | 2 | |
| **Priority** | 1 | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will allow to remove tables to a database | |
| **Preconditions** | Table must exist in the database already | |
| **Trigger** | User action | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager invokes remove table function  **Step 3:**Manager enters table ID  **Step 4:**Manager confirms that the table is to be removed | **Step 2:**System will display UI  **Step 5:**System validates the data entered   * All the data must be entered * Seats amount must be numeric * Table ID must be a number and greater than zero * Table status must be set * Table must exist in the database already   **Step 6:**  Table is removed from the Tables file |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager invokes add table function  **Step 3:**Manager enters table details | **Step 2:**System will display UI  **Step 4:**Invalid data is entered  **Step 5:**System displays error message |
| **Conclusions** | Table is being removed from the database | |
| **Post conditions** | Table cannot be accessed | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Reservations

This module will allow to manage reservations by making and cancelling them. Also two functions of the module will serve as triggers for beginning and completing reservation.

### Book reservation

This function will require Guest or Manager interaction to book a reservation

Manager

<<includes>>

Guest

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Book reservation** | |
| **Use Case Id** | 3 | |
| **Priority** | 1 | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Guest | |
| **Description** | This function will book a table for early or late time | |
| **Preconditions** | Table must not be booked for this date and time already | |
| **Trigger** | User action | |
| **Expected Scenario** | **Actor Action** | **System Response** |
| * Enter Date, number of guests * Find available tables in tables file * Assign table * Get name and phone number * Save details in reservations file | **Step 1:**Actor invokes book reservation function  **Step 3:**Details are being entered   * Number of guests * Date of arrival * Phone number and email address must be entered * Dietary/Sitting preferences(optional)   **Step 5:**Actor confirms details | **Step 2:**Booking UI is displayed  **Step 4:**System validates data entered   * Chosen table must not be booked for this date and time * Phone number and email must be in appropriate format * Number of guests must be a numerical value * Date and hour of arrival must be in appropriate format * If preferences have been entered, they must be a text value   **Step 6:**System saves data in reservations file and sends automated email with confirmation details |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Actor invokes book reservation function  **Step 3:**Details are being entered   * Number of guests * Date of arrival * Phone number and email address must be entered   Dietary/Sitting preferences(optional) | **Step 2:**Booking UI is displayed  **Step 4:**Invalid data is being entered  **Step 5:**System displays error message |
| **Conclusions** | Table is booked | |
| **Post conditions** | Table is unavailable for booking on this date and time | |
| **Business Rules** | Table cannot be booked more than one on the same time slot | |
| **Implementation Constraints** |  | |

Manager/Guest System

Invoke Book reservation

Display UI

Enter data

Validate data

Error message

Valid?

N

Y

Save booking details

Display Confirmation Message

Continue

Reset UI

### Cancel reservation

This function will require user interaction to cancel booked reservation

Manager

Guest

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Cancel reservation** | |
| **Use Case Id** | 3 | |
| **Priority** | 1 | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Guest | |
| **Description** | This function will cancel a reservation | |
| **Preconditions** | Reservation must exist already | |
| **Trigger** | User action | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Actor invokes cancel reservation function  **Step 3:**Details are being entered   * Phone number * Email address * Date of arrival     **Step 5:**Actor confirms cancellation | **Step 2:**Booking UI is displayed  **Step 4:**System validates data entered   * Table must be booked for this date and time * Phone number and email must be correct for the reservation   **Step 6:** System removes reservation from reservations file. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Actor invokes book reservation function  **Step 3:**Details are being entered   * Phone number and email address must be entered | **Step 2:**Booking UI is displayed  **Step 4:**Invalid data is being entered  **Step 5:**System displays error message |
| **Conclusions** | Reservation is removed | |
| **Post conditions** | Table is available for booking on this date and time | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

* + 1. Check – in reservation

Guest arrival will trigger this function to confirm that booked table is occupied

<<includes>>

Guest

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Check-in reservation** | |
| **Use Case Id** | 4 | |
| **Priority** | 1 | |
| **Source** |  | |
| **Primary Business Actor** | Guest | |
| **Other Participating Actors** |  | |
| **Description** | This function will confirm guest arrival | |
| **Preconditions** | Reservation must exist already | |
| **Trigger** | Guest arrival | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Guest confirms arrival | **Step 2:**Table status is changed to ‘occupied’ in the tables file, reservation status is changed to checked in in reservations table |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Guest do not arrive | **Step 2:**Table status is changed to ‘available’ in the tables file, reservation status is changed to cancelled in reservations table |
| **Conclusions** | Reservation is removed | |
| **Post conditions** | Table status is changed to occupied | |
| **Business Rules** | Table cannot be sold on that time | |
| **Implementation Constraints** | Table cannot be sold to other guests | |

* + 1. Paying Bill

This function will trigger when guest will pay for the reservation. This will change table status to available.

<<includes>>

Guest

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Regulate tab** | |
| **Use Case Id** | 5 | |
| **Priority** | 1 | |
| **Source** | Guest | |
| **Primary Business Actor** | Guest | |
| **Other Participating Actors** |  | |
| **Description** | This function will change table status when guest regulates their tab | |
| **Preconditions** | Check-in must be triggered prior | |
| **Trigger** | Guest paying for reservation | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Guest pays for reservation | **Step 2:**System stores payment in the payments file, changes associated table status to ‘available’ in tables file |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | Table status is changed, payment is logged in the system | |
| **Post conditions** | Table status is changed, payment is logged in the system | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage reporting

This module will generate various reports for analysis

### List reservations

This function will print list of reservations for particular periods

<<includes>>

Manager

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List reservations** | |
| **Use Case Id** | 6 | |
| **Priority** | 1 | |
| **Source** | Database | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will allow to print a list of daily, monthly and yearly reservations | |
| **Preconditions** | Reservations must exist in the database | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting   * Date * Guest name * Guest phone number | **Step 2:**System displays administration menu  **Step 4:**System validates data entered   * Date must be valid * Guest name must be correct * Guest phone number must be correct   **Step 5:**System prints list |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting | **Step 2:**System displays administration menu  **Step 4:**Invalid data is entered  **Step 5:**System displays error message |
| **Conclusions** | Listing is being printed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

Sample listing

Restaurant Table Reservations

Date \_\_\_/\_\_\_/\_\_\_\_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table | Name | Phone | No Guests | Arrival time | Departure time | Status |
| 1 | John O’Sullivan | 000000000 | 2 | 1700 | 1900 | Set |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |

### Analyse income

This function will allow to print revenue for various periods of time

<<includes>>

Manager

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Analyse income** | |
| **Use Case Id** | 7 | |
| **Priority** | 1 | |
| **Source** | Database | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will allow to print summary of daily, monthly and yearly revenue | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting | **Step 2:**System displays administration menu  **Step 4:**System validates data entered   * Dates must be valid   **Step 5:**System prints list |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting | **Step 2:**System displays administration menu  **Step 4:**Invalid data is entered  **Step 5:**System displays error message |
| **Conclusions** | Report is being printed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

Sample report

* + 1. Analyse volume

This function will allow to analyse guest volume for particular periods of year

<<includes>>

Manager

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Analyse volume** | |
| **Use Case Id** | 8 | |
| **Priority** | 1 | |
| **Source** | Database | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will allow to print summary of daily, monthly and yearly guest volume | |
| **Preconditions** | Reservations must exist in the database | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting | **Step 2:**System displays administration menu  **Step 4:**System validates data entered   * Dates must be valid   **Step 5:**System prints list |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step 1:**Manager login  **Step 3:**Manager inputs data for reporting | **Step 2:**System displays administration menu  **Step 4:**Invalid data is entered  **Step 5:**System displays error message |
| **Conclusions** | Report is being printed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

Sample report

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD RestaurantSYS

Restaurant booking system

Reservation details

Guest

Invokes

## Level-1 DFD System Modules

Guest

D1

Bills file

P2

Manage Reservations

P1

Manage Tables

Reservations

File

D3

D2

Table file

P3

Admin

## Level-2 DFD (Process P1: Title)

## Level-2 DFD (Process P2: Title)

## Level-2 DFD (Process P3: Title)

# Data Model (Class Diagram)

Brief introduction……

## Class Diagram

Object Model – UML Class Diagram

Class diagram shows objects & attributes

## Relational Schema

Relational schema for the data requirements - Using ***bracket notation***

## Database Schema

A definition of the database to be implemented.

This includes primary key, foreign key and other constraints to be implemented.

# Conclusion

# Appendices

## Appendix A – Title

## Appendix B – Title

Might include:

* **Lookup / Reference tables**
* **Sample reports / Listings**