Requirements Engineering

Office Room Booking System

Submitted By: Mateusz Halada (T00205630)

Computing with Games Development

Date Submitted:

**Table of Contents**

[1. Introduction/overview 3](#_Toc526502558)

[2. Functional Components 4](#_Toc526502559)

[3. User Requirements 5](#_Toc526502560)

[3.1. Rooms will manage Rooms 5](#_Toc526502561)

[3.2. Booking will process reservations 5](#_Toc526502562)

[3.3. Admin will perform administrative reporting 5](#_Toc526502563)

[4. System Requirements 6](#_Toc526502564)

[4.1. System Level Use Case Diagram 6](#_Toc526502565)

w[4.2. Manage Office Rooms 6](#_Toc526502566)

[4.2.1. Add Rate 6](#_Toc526502567)

[4.2.2. Update Rate 6](#_Toc526502568)

[4.2.3. List Rooms 6](#_Toc526502569)

[4.3. Make Bookings 6](#_Toc526502570)

[4.3.1. Make Booking 6](#_Toc526502571)

[4.3.2. Cancel Booking 6](#_Toc526502572)

[4.3.3. Update Booking 6](#_Toc526502572)

[4.4. Perform Admin 6](#_Toc526502573)

[4.4.1. Revenue Analysis 6](#_Toc526502574)

[4.4.2. Office Revenue Analysiss 6](#_Toc526502575)

[5. System Model 7](#_Toc526502576)

[5.1. Level-0 DFD 7](#_Toc526502577)

[5.2. Level-1 DFD 7](#_Toc526502578)

[5.3. Level-2 DFD (Process P1: Title) 7](#_Toc526502579)

[5.4. Level-2 DFD (Process P2: Title) 7](#_Toc526502580)

[5.5. Level-2 DFD (Process P3: Title) 7](#_Toc526502581)

[6. Data Model (Class Diagram) 8](#_Toc526502582)

[6.1. Class Diagram 8](#_Toc526502583)

[6.2. Relational Schema 8](#_Toc526502584)

[6.3. Database Schema 8](#_Toc526502585)

[7. Conclusion 9](#_Toc526502586)

[8. Appendices 10](#_Toc526502587)

[8.1. Appendix A – Title 10](#_Toc526502588)

[8.2. Appendix B – Title 10](#_Toc526502589)

# Introduction/overview

An office room booking system for a business which rents out office rooms.

The system has a list of rooms available and bookings for dates.

The system can add rooms with dates available for booking.

The system can make bookings for a specific amount of days.

The system can update availability of the rooms along with daily rates.

The system can cancel bookings.

The system has a list of office rooms with their ids, numbers and floors, and price bracket.  
An admin can change the data.

# Functional Components

This section presets the functional components of the propsoed software system.

There are 3 main componets in the proposed system

# User Requirements

This section describes the user requirements (functional components) as high-level abstract statements.

## Rooms will manage Rooms

* + 1. Add Room will add an office room.
    2. Update Room will update the office room.
    3. List Rooms will find available rooms.

## Bookings will process reservations

* + 1. Make Booking will create a booking.
    2. Update Booking will update booking.
    3. Cancel Booking will cancel booking.

## Admin will perform administrative reporting

* + 1. Revenue Analysis will produce a revenue analysis.
    2. Office Analysis will produce a revenue analysis for one office room.

# System Requirements

Office Room Rental System consists of three main modules.

The rooms module provides functions to add rate, update rate, add a room and check for available rooms in list rooms.

The booking module provides functions to add booking, update booking, and cancel booking.

The admin module provides functions to analyse total annual revenue and single room annual revenue.

## System Level Use Case Diagram

The following system level use case diagram illustrates the high-level system requirements.

Manager

Business/Individual

Receptionist

## Manage Office Rooms

This module provides functions to add rate, update rate, add room, update room and list room availability.

### Add Rate This function adds a rate for a room type to the database.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **AddRate** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function adds a rate into the database. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Manager Invokes Add rate function.  **Step 2:** Retrieve Room Rates from the rates and load on UI in alphabetical order.  **Step 3:**  The Manager enters the required data:   * OfficeNo * OfficeRate   **Step 4:** The Manager confirms rate is to be added. | **Step 5:** The system validates the data entered:   * OfficeNo must be entered * OfficeNo must be numeric size 3 * OfficeNo must not already exist * OfficeRate Must Be Entered   **Step 6:**  The system saves the details in the Rooms File.  **Step 7:**  The system displays a confirmation message.  **Step 8:** Reset UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Office Already Exists** |  | **Step 5:**  OfficeNo already exists  **Step 6:**  Display appropriate error message  **Step 7:** Return to step 3 |
| **Conclusions** | The Room has been added to the database. | |
| **Post conditions** | The office added can now be booked. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update Rate

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **UpdateRate** | |
| **Use Case Id** | 2 | |
| **Priority** | 2 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function updates a rate in the database. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Manager Invokes Update rate function.  **Step 2:** Manager Enters old OfficeNo  **Step 3:**  The Manager enters the new data:   * OfficeNo * OfficeRate   **Step 4:** The Manager confirms rate is to be updated. | **Step 5:** The system validates the data entered:   * Old OfficeNo entered must match the OfficeNo in the database * OfficeNo must be entered * OfficeNo must be numeric size 3 * OfficeNo must not already exist   **Step 6:**  The system saves the details in the Rooms File.  **Step 7:**  The system displays a confirmation message.  **Step 8:** Reset UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **OfficeNo not recognised** |  | **Step 5:**  OfficeNo does not match any entry in the database  **Step 6:**  Display appropriate error message  **Step 7:** Return to step 2 |
| **Conclusions** | The Room has been updated in the database. | |
| **Post conditions** | The office now has new details and credentials. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### List Rooms

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Rooms** | |
| **Use Case Id** | 3 | |
| **Priority** | 3 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function lists rooms within the database. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Manager calls the list rooms function.  **Step 2:** Retrieve Room List from the rates and load on UI in alphabetical order.  **Step 3:**  Manager enters search filters. | **Step 4:** Reset UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Room List has been pulled from the database and displayed. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Make Booking

### Make Booking

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **MakeBooking** | |
| **Use Case Id** | 4 | |
| **Priority** | 4 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function makes a booking for a room in the database. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Manager invokes the Make Booking function.  **Step 2:** Manager inputs room number, date from, and date to  **Step 3:** Manager confirms he wants to make booking. | **Step 4:** The system validates the data entered:   * OfficeNo must match in the database. * OfficeNo must be entered * OfficeNo must be numeric size 3 * OfficeNo must not already be booked. * Date From must not already be booked in that room * The duration between date from and date to must not already be booked.   **Step 5:**  The system saves the details in the Rooms File.  **Step 6:**  The system displays a confirmation message.  **Step 7:** Reset UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Cancel Booking

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Room** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** |  | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  |  |  |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### 4.3.3. Update Booking

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Room** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function adds a rate into the database. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  |  |  |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Perform Admin

### Analyse Revenue

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Room** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** |  | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  |  |  |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Analyse Office

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Room** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** |  | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  |  |  |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

## Level-1 DFD

## 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**