Requirements Engineering

Desk Rental System

Submitted By: Eleonora Blazhko (T00229088)

Computing with Software Development

Date Submitted: 08/12/2023

**Table of Contents**

[1. Introduction/overview 4](#_Toc152948018)

[2. Functional Components 5](#_Toc152948019)

[3. User Requirements 6](#_Toc152948020)

[3.1. DeskRentalSYS will manage Desks 6](#_Toc152948021)

[3.2. DeskRentalSYS will manage Customers 6](#_Toc152948022)

[3.3. DeskRentalSYS will manage Rental System 6](#_Toc152948023)

[3.4. DeskRentalSYS will perform administrative reporting 6](#_Toc152948024)

[4. System Requirements 7](#_Toc152948025)

[4.1. System Level Use Case Diagram 7](#_Toc152948026)

[4.2. Desks Management 8](#_Toc152948027)

[4.2.1. Add Desk 8](#_Toc152948028)

[4.2.2. Update Desk 10](#_Toc152948029)

[4.2.3. Remove Desk 12](#_Toc152948030)

[4.3. Manage Customers 14](#_Toc152948031)

[4.3.1. Register Customer 14](#_Toc152948032)

[4.3.2. Update Customer 16](#_Toc152948033)

[4.4. Process Bookings 22](#_Toc152948034)

[4.4.1. Book Desk 22](#_Toc152948035)

[4.4.2. Cancel Booking 25](#_Toc152948036)

[4.5. Process Data Analysis 27](#_Toc152948037)

[4.5.1. Analyse Customer Preferences 27](#_Toc152948038)

[Analyse Customer Preferences 27](#_Toc152948039)

[Analyse Yearly Revenue 29](#_Toc152948040)

[Figure 2 Sample Analyse Yearly Revenue 30](#_Toc152948041)

[5. System Model 31](#_Toc152948042)

[5.1. Level-0 DFD 31](#_Toc152948043)

[5.2. Level-1 DFD 32](#_Toc152948044)

[5.3. Level-2 DFD - (Process P1: Manage Desks) 33](#_Toc152948045)

[5.4. Level-2 DFD – (Process P2: Manage Customers) 34](#_Toc152948046)

[5.5. Level-2 DFD – (Process P3: Process Bookings) 35](#_Toc152948047)

[5.6. Level-2 DFD – (Process P4: Process Data Analysis) 36](#_Toc152948048)

[6. Data Model (Class Diagram) 37](#_Toc152948049)

[6.1. Class Diagram 37](#_Toc152948050)

[6.2. Relational Schema 38](#_Toc152948051)

[6.3. Database Schema 39](#_Toc152948052)

[7. Conclusion 41](#_Toc152948053)

# Introduction/overview

The Desk Rental System (DeskRentalSYS) stands as an advanced platform meticulously designed to optimize the operational facets of a desk rental service. Tailored to accommodate the needs of both customers and desk management personnel, DeskRentalSYS offers a seamlessly efficient environment for desk reservations, comprehensive customer and desk management, and insightful data analysis.

At its essence, DeskRentalSYS embodies pivotal functional components, encompassing desk and customer management, booking system operations, and meticulous administrative reporting. Customers have the capability to initiate reservations, while staff possesses the authority to cancel reservations when necessary. On the management front, staff members are empowered to add, update, and remove desks, register or deregister customers, and maintain up-to-date customer information.

The system's operational intricacies are meticulously outlined in the system requirements section, elucidated by a system-level use case diagram, and specific use cases tailored for desk and customer management, booking processes, and data analysis. The system model, as articulated through Data Flow Diagrams (DFDs), offers a transparent depiction of the fluidity of information and processes within DeskRentalSYS across varying hierarchical tiers.

The data model, encapsulated within a Class Diagram, furnishes a visual comprehension of the entities and interrelationships intrinsic to the system, accentuating the foundational data structure. This includes comprehensive details pertaining to desks, customers, rentals, and rates — all indispensable for the seamless operation of the desk rental service.

DeskRentalSYS stands as an exhaustive and refined solution, augmenting the operational efficiency of desk rental processes for both customers and management personnel. This document serves as an authoritative guide for stakeholders, unveiling the nuanced intricacies of the system with precision and clarity.

# Functional Components

# User Requirements

## DeskRentalSYS will manage Desks

* + 1. DeskRentalSYS will add a desk to the system
    2. DeskRentalSYS will update a desk
    3. DeskRentalSYS will remove a desk from the system

## DeskRentalSYS will manage Customers

* + 1. DeskRentalSYS will enable the addition of a new customer
    2. DeskRentalSYS will support updating customer details
    3. DeskRentalSYS will provide the option to delete a customer profile

## DeskRentalSYS will manage Rental System

* + 1. DeskRentalSYS will allow customers to book desks for specific periods
    2. DeskRentalSYS will enable customers to cancel a booking

## DeskRentalSYS will perform administrative reporting

* + 1. DeskRentalSYS will provide a yearly revenue analysis
    2. DeskRentalSYS will provide insights into customer booking behaviour and preferences

# System Requirements

The system requirements of the system are presented in this section of the document. Include a brief outline the top-level modules in your system

## System Level Use Case Diagram

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

Customer

Manager

## Desks Management

This module provides functions to add new desks, update existing desks, and remove desks.

### Add Desk

This use case describes the process of adding a new desk to the Desk Rental System. A desk includes attributes such as desk type, description, and rate.

<<includes>>

<<extends>>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Add Desk | |
| **Use Case Id** | 001 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case describes the process of adding a new desk to the Desk Rental System. A desk includes attributes such as desk type, description, and rate. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The Manager initiates the "Add Desk" function.  **Step 4:** The Manager enters the required data:   * DeskType * DeskNumber * Rate * Description   **Step 5:** The Manager confirms. | **Step 2:** The system retrieves information from DeskTypes File and loads on UI.  **Step 3:** The system displays the UI.  **Step 6:** The system validates the entered data:  • All required fields must be entered.  • The Desk Type must be valid.  • The description of a desk must not consist of all numeric characters.  • The Rate must be a numeric value greater than 0.  **Step 7:** The system assigns a unique ID to the desk.  **Step 8:** The system saves the desk’s information in the Desks File:  • Desk\_ID  • Desk Number  • Desk Type  • Description  • Rate  • Status  **Step 9:** The system displays a confirmation message.  **Step 10:** The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Invalid Data Input** | **Step 6**: Invalid data found. | **Step 7**: The system displays an appropriate error message.  **Step 8**: Return to **Step 3**. |
| **Conclusions** | A new desk has been successfully added to the Desks File. | |
| **Post conditions** | The desk can now be booked. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update Desk

This function updates the details of a desk. A desk includes attributes such as the desk type, description, and rate.

<<extends>>

<<includes>>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Desk | |
| **Use Case Id** | 002 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function updates the details of a desk. A desk includes attributes such as the desk type, description, and rate. | |
| **Preconditions** | * The user has the necessary permissions to update desk details. | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The Manager initiates the "Update Desk" function.  **Step 4:** The Manager enters the required data:   * DeskNumber   **Step 7:** The Manager updates the required data:  • Desk Number  • Desk Type  • Description  • Rate  • Status  **Step 8:** The Manager confirms. | **Step 2:** The system retrieves Desk details from Desks File and loads on UI.  **Step 3:** The System displays the UI.  **Step 5:** The system validates the entered data:  • DeskNumber must be entered.  • DeskNumber must be valid.  • Desk Number must consist of all numeric characters.  **Step 6:** The System displays the UI.  **Step 9:** The system validates the entered data:  • All required fields must be entered.  • The Desk Type must be valid.  • The description of a desk must not consist of all numeric characters.  • The Rate must be a numeric value greater than 0.  **Step 10:** The system updates the desk’s information in the Desks File:  • Desk Number  • Desk Type  • Description  • Rate  • Status  **Step 11:** The system displays a confirmation message.  **Step 12:** The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Desk Number not found** | **Step 4**: Invalid data entered | **Step 5**: The system displays an error message.  **Step 6**: Return to **Step 4**. |
| **Invalid Data Input** | **Step 9:** Invalid data entered | **Step 10**: The system displays an appropriate error message.  **Step 11**: Return to **Step 7**. |
| **Conclusions** | The details of the desk have been successfully updated. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Remove Desk

This use case removes a desk. A desk can be removed when it is no longer available or needed.

<<extends>>

<<includes>>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Remove Desk | |
| **Use Case Id** | 003 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case removes a desk. A desk can be removed when it is no longer available or needed. | |
| **Preconditions** | * The user is logged into the Desk Rental System. * The user has the necessary permissions to remove desks. | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The Manager initiates the "Remove Desk" function.  **Step 4:** The Manager enters the required data:   * DeskNumber   **Step 7:** The Manager confirms. | **Step 2:** The system retrieves Desk details from Desks File and loads on UI.  **Step 3:** The System displays the UI.  **Step 5:** The system validates the entered data:  • DeskNumber must be entered.  • DeskNumber must be valid.  • Desk Number must consist of all numeric characters.  **Step 6:** The system displays the UI.  **Step 8:** Desk is set to Removed “R” in Desks File.  **Step 9:** The system displays a confirmation message.  **Step 11:** The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Desk Number not found** | **Step 4:** Invalid data entered | **Step 5**: The system displays an error message.  **Step 6**: Return to **Step 4**. |
| **Conclusions** | The details of the desk have been successfully updated. | |
| **Post conditions** | The Desk is no longer available for booking. | |
| **Business Rules** | * A desk can only be removed if it is available (not currently booked) and there are no outstanding issues associated with it. | |
| **Implementation Constraints** |  | |

## Manage Customers

This module provides functions to register a new customer, update customers details and deregister customers.

### Register Customer

This use case registers a new customer. Registration involves capturing customer information to enable booking and communication.

<<includes>>

<<extends>>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Register Customer | |
| **Use Case Id** | 010 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case registers a new customer. Registration involves capturing customer information to enable booking and communication. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The Manager initiates the "Register Customer" function**.**  **Step 3:** The Manager enters the customer's details:  • Forename  • Surname  • Email  • Phone  • Town  • Address Line 1  • Address Line 2  • County  • Eircode  **Step 4:** The Manager confirms. | **Step 2:** The system displays the UI.  **Step 5:** The system validates the entered data:  • All required fields must be entered.  • The email must be in a valid format.  • The phone number must be in a valid format.  • The Eircode must be in a valid format.  • Forename must not be numeric.  • Surname must not be numeric.  • Town must not be numeric.  • County must not be numeric.  **Step 6:** The system generates a value Customer\_ID.  **Step 7:** Set customer status to active (‘A’).  **Step 8:** The system saves Customer's information in the Customers File:  • Customer\_ID  • Fname  • Sname  • Email  • Phone  • Town  • Address1  • Address2  • County  • Eircode  • Status  **Step 9:** The system displays a confirmation message.  **Step 10**: The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Invalid Data Input** | **Step 5:** Invalid data entered | **Step 6**: The system displays an appropriate error message.  **Step 7**: Return to Step 3. |
| **Conclusions** | A new customer has been successfully registered in the system. | |
| **Post conditions** | The customer is now able to proceed with a booking. | |
| **Business Rules** | The System cannot register a customer if he/she is already in the system. | |
| **Implementation Constraints** |  | |

### Update Customer

This use case updates a customer.

<<extends>>

<<includes>>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Customer | |
| **Use Case Id** | 011 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function allows to updates a customer information. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the "Update Customer" function**.**  **Step 4:** The Assistant enters the required data:  • Surname  **Step 8:** The Assistant updates the customer's details:  • Fname  • Sname  • Email  • Phone  • Town  • Address1  • Address2  • County  • Eircode | **Step 2:** The system retrieves Surname from Customers File and loads on UI.  **Step 3:** The system displays the UI.  **Step 5:** The system validates the entered data:  • The field must be entered.  • The Surname must be valid.  **Step 6:** The system retrieves the information from Customers File and loads on UI for updating.  **Step 7:** The system updates the UI.  **Step 9:** The system validates the entered data:  • All required fields must be entered.  • The email must be in a valid format.  • The phone number must be in a valid format.  • The Eircode must be in a valid format.  • Forename must not be numeric.  • Surname must not be numeric.  • Town must not be numeric.  • County must not be numeric.  **Step 9:** The system updates the customer’s information in the Customers File:  • Fname  • Sname  • Email  • Phone  • Town  • Address1  • Address2  • County  • Eircode  **Step 10:** The system displays a confirmation message.  **Step 11**: The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Invalid Data Input** | **Step 5:** Invalid data entered | **Step 6**: The system displays an appropriate error message.  **Step 7**: Return to Step 3. |
| **Customer Details Validation failed** | **Step 9:** Invalid data entered. | **Step 10**: The system displays an appropriate error message.  **Step 11:** Return to **Step 8**. |
| **Conclusions** | A customer details has been successfully updated in the system. | |
| **Post conditions** | The customer is now able to proceed with a booking. | |
| **Business Rules** | The system can update only Active Customers. | |
| **Implementation Constraints** |  | |

* + 1. Deregister Customer

This function allows to deregister a customer details.

<<includes>>

Manager

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Deregister Customer | |
| **Use Case Id** | 012 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function deregisters a customer. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the "Update Customer" function**.**  **Step 4:** The Assistant enters the required data:  • Surname  **Step 8:** The Assistant clicks removal button. | **Step 2:** The system retrieves Surname from Customers File and loads on UI.  **Step 3:** The system displays the UI.  **Step 5:** The system validates the entered data:  • The field must be entered.  • The Surname must be valid.  **Step 6:** The system retrieves customer details from Customers File and displays on UI for viewing only.  **Step 7:** The system updates the UI.  **Step 9:** The system sets the status to deleted (‘D’) in the Customers File.  **Step 10:** The system displays a confirmation message.  **Step 11:** The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Invalid Data Input** | **Step 5:** Invalid data entered. | **Step 6**: The system displays an appropriate error message.  **Step 7:** Return to **Step 4**. |
| **Conclusions** | The details of a customer have been set to deleted. | |
| **Post conditions** | The Customer has no longer access for bookings. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

|  |  |
| --- | --- |
| **Manager** | **System** |
|  | [Y]  [N]  Valid?  [Y]  [N]  Valid? |

## Process Bookings

This module includes functions for customers to book and cancel bookings.

### Book Desk

This use case involves a customer booking a desk within the DeskRentalSYS system. Booking includes selecting a desk, specifying booking details, and making a reservation for a specific period.

Customer

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Book Desk | |
| **Use Case Id** | 021 | |
| **Priority** | High | |
| **Source** | Assistant | |
| **Primary Business Actor** | Assistant | |
| **Other Participating Actors** | Customer | |
| **Description** | This use case involves a customer booking a desk within the DeskRentalSYS system. Booking includes selecting a desk, specifying booking details, and making a reservation for a specific period. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the "Book Desk" function.  **Step 4:** The user selects the required data:   * DeskType * DateFrom * DateTo   **Step 7:** The user selects available desk.  **Step 9:** The user enters the required data:   * Fname * Sname * Email * Phone | **Step 2:** The system retrieves data from DeskTypes File and loads on UI.  **Step 3:** The system displays the UI.  **Step 5:** The system retrieves information about available desks from the Desks File, checking whether there are free on the chosen dates and loads on UI.  **Step 6:** The system updates UI.  **Step 8:** The system calculates TotalCost (No Days \* Rate) and displays on the UI.  **Step 10:** The system validates the entered data:  • All required fields must be entered.  • The email must be in a valid format.  • The phone number must be in a valid format.  • Forename must not be numeric.  • Surname must not be numeric.  **Step 11:** The system generates a value Booking\_ID.  **Step 15:** The system saves the Booking information in the Booking File:   * Booking\_ID * DeskType * DeskNumber * DateFrom * DateTo * Fname * Sname * Email * Phone * TotalCost * Status   **Step 16:** The system displays a confirmation message.  **Step 17:** The system resets the UI. |
| **Alternate Scenarios** | **Customer** | **System** |
| **Entered data is missing or empty** | **Step 9:** Invalid data entered. | **Step 10**: The system displays an appropriate error message.  **Step 11:** Return to **Step 9**. |
| **Conclusions** | The Customer has successfully booked a desk for the specified period. | |
| **Post conditions** | The desk cannot be booked on these dates by other  Customers. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Cancel Booking

This use case involves a cancelation of customer booking a desk within the DeskRentalSYS system.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Cancel Booking | |
| **Use Case Id** | 022 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case cancels a customer booking. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the "Cancel Booking" function.  **Step 3:** The user enters the required data:  • Surname  **Step 6:**  The user then selects the booking to be cancelled. | **Step 2:** The system displays UI.  **Step 4:** The system validates the entered data:  • The field must be entered.  • The Surname must be valid.  • The Surname must be in the Customer File.  **Step 5:** The system retrieves all future bookings for customer from Bookings file and loads them on UI.  **Step 7:** The system Deletes the booking from the Booking File.  **Step 8:** The system displays a confirmation message.  **Step 9:** The system resets the UI. |
| **Alternate Scenarios** | **Manager** | **System** |
| **Customer Details Validation failed** | **Step 4:** Invalid data entered. | **Step 5**: The system displays an appropriate error message.  **Step 6:** Return to **Step 3**. |
| **Conclusions** | The booking has been successfully cancelled. | |
| **Post conditions** | A desk is available for new bookings on the dates it was booked. | |
| **Business Rules** | Bookings can only be cancelled if they are within the allowed cancellation time frame (2 days). | |
| **Implementation Constraints** |  | |

## Process Data Analysis

### Analyse Customer Preferences

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Analyse Customer Preferences | |
| **Use Case Id** | 030 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case analysing customer preferences. The system will provide insights into the types of desks preferred by customers based on their booking history. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the " Analyse Customer Preferences" function.  **Step 4:** User selects year to be analysed. | **Step 2:**  The system loads current year and three previous years on to UI.  **Step 3:** Display UI.  **Step 5:** The system retrieves data from Bookings file for the chosen year.  **Step 6:** The system groups the booking data by DeskTypes for each month.  **Step 7:** The system calculates the number of bookings for a desk type in a month.  **Step 8:**  The system displays UI with pie chart. |
| **Alternate Scenarios** | **Actor** | **System** |
|  |  |  |
| **Conclusions** | Analyse Customer Preferences has been successfully generated. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

A blue pie chart with white text

Description automatically generated

Figure 1 Sample Analyse Customer Preferences

4.5.2. Analyze Yearly Revenue

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Analyse Yearly Revenue | |
| **Use Case Id** | 031 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function allows to analyse Yearly Revenue. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** The user initiates the "Analyse Yearly Revenue" function.  **Step 4:** User selects year to be analysed. | **Step 2:**  The system loads current year and three previous years on to UI.  **Step 3:** Display UI.  **Step 5:** The system retrieves data from Bookings and Desks files for the chosen year.  **Step 6:** The system determines the total income (Cost) for each month of the selected year.  **Step 7:**  The system displays UI with bar chart. |
| **Alternate Scenarios** | **Actor** | **System** |
|  |  |  |
| **Conclusions** | The system has successfully generated a yearly revenue analysis report. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

Figure 2 Sample Analyse Yearly Revenue

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Booking Details

Customer

DeskRentalSYS

Invoice

## Level-1 DFD

Desks Details

Desks Details

Process Bookings

P3

Desks Details

Customer Details

Booking Details

Manage Desks

P1

D1

Desks File

D3

Bookings File

Customer

Customers Details

Bookings Details

Desk Type Details

Process Data Analysis

P4

D2

Customers File

Manage Customers

P2

Customers Details

## Level-2 DFD - (Process P1: Manage Desks)

Desk Details

New Desk Status

Remove Desk

P1.3

Add Desk

P1.1

D1

Desks File

Current Desk Details

Current Desk Details

New Desk Details

Update Desk

P1.2

## Level-2 DFD – (Process P2: Manage Customers)

New Customer Status

Deregister Customer

P2.3

Register Customer

P2.1

Customer Details

D2

Customers File

Current Customer Details

Customer Details

New Customer Details

Current Customer Details

Customer

Update Customer

P2.2

Customer Details

## Level-2 DFD – (Process P3: Process Bookings)

Order Details

Customer

Order Details

D3

D2

Booking Details

Desks File

Booking Details

Reservation No

Customer Details

Customers File

Cancel Booking

P3.2

Reservation No

Book Desk

P3.1

Bookings File

Booking Details

Updated Booking Details

Desk Type Details

D1

## Level-2 DFD – (Process P4: Process Data Analysis)

Booking Details

Booking Details

Booking Details

Analyse Yearly Revenue

P4.2

Analyse Customer Preferences

P4.1

D3

D1

Desk Details

Desks File

Bookings File

# Data Model (Class Diagram)

This data model includes a UML class diagram, a relational schema, and a database scheme for the DeskRentalSYS.

## Class Diagram

Customer

- Customer\_ID\*: int

- FirstName: String

- LastName: String

- Email: String

- Phone: varchar2archar2

-Town: String

- Address: String

- County: String

- Eircode: String

- Status: char

Desk Type

- TypeCode\*: int

- Description: varchar2

0…\*

🡪

has a

1

Desk

- Desk\_ID\*: int

- DeskNumber: int

- Description: varchar2

- Rate: int

- Status: char

1

has a

🡪

Booking

- Booking\_ID\*: int

- BookDate: Date

- ArrivalDate: Date

- DepatureDate: Date

- TotalCost: int

0…\*

🡪

0…\*

1

contains

has a

1…\*

1

BookedDesk

- DeskCost: int

🡪

## Relational Schema

Customers (Customer\_ID, FirstName, LastName, Email, Phone, Town, Address, County, Eircode, Status)

DeskTypes(TypeCode, Description)

Desk (Desk\_ID, DeskNumber, Description, DeskType, Rate, Status)

Bookings (Booking\_ID, Customer\_ID, BookDate, ArrivalDate, DepatureDate, Cost)

BookedDesks (Booking\_ID, Desk\_ID, DeskCost)

## Database Schema

A definition of the database to be implemented.

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

**Relation Desks**

**Attributes:**

Desk\_ID: int (4) NOT NULL,

DeskNumber: int (4) NOT NULL,

Description: varchar2(40),

Rate: number(4,2) NOT NULL,

Status: char(1) NOT NULL

**Primary Key:** Desk\_ID

**Relation Customers**

**Attributes:**

Customer\_ID: int (4) NOT NULL,

FirstName: String (15) NOT NULL,

LastName: String (15) NOT NULL,

Email: String (30) NOT NULL,

Phone: varchar2(15) NOT NULL,

Town: String(20) NOT NULL,

Address: String(30) NOT NULL,

County: Strig(10) NOT NULL,

Eircode: String(10) NOT NULL,

Status: char (1) NOT NULL

**Primary Key:** Customer\_ID

**Relation Bookings**

**Attributes:**

Booking\_ID: int (4) NOT NULL,

Customer\_ID: int (4) NOT NULL,

BookDate: Date NOT NULL,

ArrivalDate: Date NOT NULL,

DepatureDate: Date NOT NULL,

TotalCost: int(4,2)NOT NULL

**Primary Key:** Booking\_ID

**Foreign Key:** Customer\_ID **References** Customers

**Relation BookedDesks**

**Attributes:**

Booking\_ID: int (4) NOT NULL,

Desk\_ID: int (4) NOT NULL,

DeskCost: int (5,2) NOT NULL

**Foreign Key:** Booking\_ID **References** Bookings

**Foreign Key:** Desk\_ID **References** Desks

**Relation DeskTypes**

**Attributes:**

TypeCode: int (4) NOT NULL,

Description: varchar2(40)

**Primary Key:** TypeCode

# Conclusion

The formulation of this requirements document has outlined the essential processes and functionalities integral to the development of the Desk Rental System. Through the comprehensive creation of detailed use case narratives for each function, the blueprint for user interfaces, data presentation, and data storage has been established. The use case narratives played a pivotal role in uncovering the data requirements essential for system functionality, serving as a foundational guide for constructing the class diagram, relational schema, and database schema specific to the Desk Rental System.

Simultaneously, a prototype of the system was developed in tandem with the requirements document. This parallel approach proved immensely beneficial, providing a tangible glimpse into the prospective operation of the system. Any user issues identified during prototype testing significantly influenced the refinement of associated functions documented in the requirements.

In conclusion, the symbiotic development of the requirements document and system prototype has been instrumental in shaping the trajectory of the Desk Rental System. The insights gained from detailed use case narratives and the iterative development process ensure a foundation for the system's construction and future operationalization.