*Department of Information Systems*

***Systems Design & Development***

**



**Table Of Contents**

[**1.**](#_heading=h.gjdgxs) **Introduction 1**

[1.1.](#_heading=h.30j0zll) Overview of Specification 1

[1.2.](#_heading=h.1fob9te) Context & Scope of System Specification 1

[1.3.](#_heading=h.3znysh7) Design Assumptions & Constraints 1

[**2.**](#_heading=h.2et92p0) **User Interface & Dialogue Design 1**

[2.1.](#_heading=h.3dy6vkm) Interface Flow Diagrams 2

[2.2.](#_heading=h.1t3h5sf) Screen Standards 2

[2.3.](#_heading=h.4d34og8) Detailed Screen Layout 2

[**3.**](#_heading=h.2s8eyo1) **Design Sequence Diagrams 2**

[3.1.](#_heading=h.17dp8vu) Design Sequence Diagram 1 2

[3.2.](#_heading=h.3rdcrjn) Design Sequence Diagram 2 3

[**4.**](#_heading=h.26in1rg) **Design Class Diagrams 3**

[**5.**](#_heading=h.lnxbz9) **Entity Relationship Diagram 3**

[**6.**](#_heading=h.35nkun2) **Report Design 4**

[6.1.](#_heading=h.1ksv4uv) Report 1 5

[*6.1.1.*](#_heading=h.44sinio) *Detailed Output Requirements 5*

[*6.1.2.*](#_heading=h.2jxsxqh) *Report Layout 5*

[6.2.](#_heading=h.z337ya) Report 2 5

[*6.2.1.*](#_heading=h.3j2qqm3) *Detailed Output Requirements 5*

[*6.2.2.*](#_heading=h.1y810tw) *Report Layout 5*

[**7.**](#_heading=h.4i7ojhp) **Input-Output Standards & Controls 6**

[7.1.](#_heading=h.1ci93xb) Formalised Outputs: 6

[7.2.](#_heading=h.3whwml4) Built-In Validation to Ensure Requirements are Met 6

[7.3.](#_heading=h.2bn6wsx) Input Integrity Controls 6

[7.4.](#_heading=h.qsh70q) Output Integrity Controls 6

[**8.**](#_heading=h.3as4poj) **Implementation Plan 6**

[**9.**](#_heading=h.1pxezwc) **Test Plan 6**

[9.1.](#_heading=h.49x2ik5) Test Environment 7

[9.2.](#_heading=h.2p2csry) Test Items 7

[9.3.](#_heading=h.147n2zr) Test Approaches 7

[9.4.](#_heading=h.3o7alnk) Problem Tracking (Test Cases) 8

[9.5.](#_heading=h.23ckvvd) Test Schedule 8

# Introduction

## Overview of Specification

The **Phumla Kamnandi Hotel** group was formed in 2020 and aims to manage a portfolio of small hotels in South Africa, the group currently has a total of 30 hotels and each hotel runs independently.

There are currently no hardware or software standards across all the hotels managed by the group, and most hotels have old operating systems which cause havoc in managing bookings. The project was given rise by the fact that the company wants to create a new computer system and implement the same system in all 30 existing hotels. The system hopes to achieve the following:

1. Best practice business process that will be enforced by the software.

2. Standard image I appearance of accounts,

3. Standardised management and financial reporting across the group, and easier consolidation of information at hotel level and at group level

4. Reduction in the total cost of ownership (TCO) hardware, software, and training across the group.

**The analysis phase** of the project was concluded 6 months ago and is now at the design and development phase of the project. This part of the project aims to design and develop the system as per the requirements discussed in the Phumla Kamnandi Hotel Group. This document contains the following: context and scope of the system, design assumptions and constraints, user interface and dialogue design, design sequence diagrams, design class diagrams, entity-relationship diagram, report design, input-output standards and controls, and finally the implementation plan.

.

.

## Context & Scope of System Specification

For this project, the Waterfall methodology is being followed. It is a methodology that is made up of six phases having defined start and endpoints, where each phase must be completed before the next phase can begin.

Previously in the INF2009F project, we completed the first two phases:

1. **Initiation** – A Business Case was produced, which included the business problems, opportunities, objectives, scope, candidate solutions, feasibility, risks, and project planning.

Through this, a thorough analysis was made on the business processes and other existing systems and how they interact with one another

2. **Analysis** –For this, a User Requirements Specification was developed and focuses on satisfying the end-user.

*Currently, the project is undergoing the following three phases:*

3. **System design** – Design artifacts were produced for this phase, namely Dialog design, Sequence diagram, database design, Report Design, etc.

4. **Implementation**- Coding of the proposed and the designed system took place in this phase, a C# solution was produced.

5. **Testing** – Detailed design of system and subsystem inputs and outputs relative to the user were provided for this phase.

Many methods were implemented to obtain the user requirements. Reports of previous projects and currently existing projects were reviewed to avoid repeating mistakes from previously implemented projects and use that knowledge to improve the existing ones. Meetings were held with employees who use the system at Phumla Kamnandi Hotel group and there was frequent communication with the manager. This interaction allowed a better understanding of what the systems’ issues are and how the workers, management, and clients(feedback) hope it is improved.

**Objectives and requirements**

The first phase of the Phumla Kamnandi Hotels computerization project will be completed in several iterations. One of this project’s requirements is to complete a single iteration which is to construct and implement the Guest Booking function.

The “phone in” booking system has the following requirements and needs to provide the following functionality:

The system should allow the receptionist to change an existing booking and adjust hotel occupancy details accordingly

1. The system should allow the receptionist to cancel a current booking and adjust hotel occupancy details.

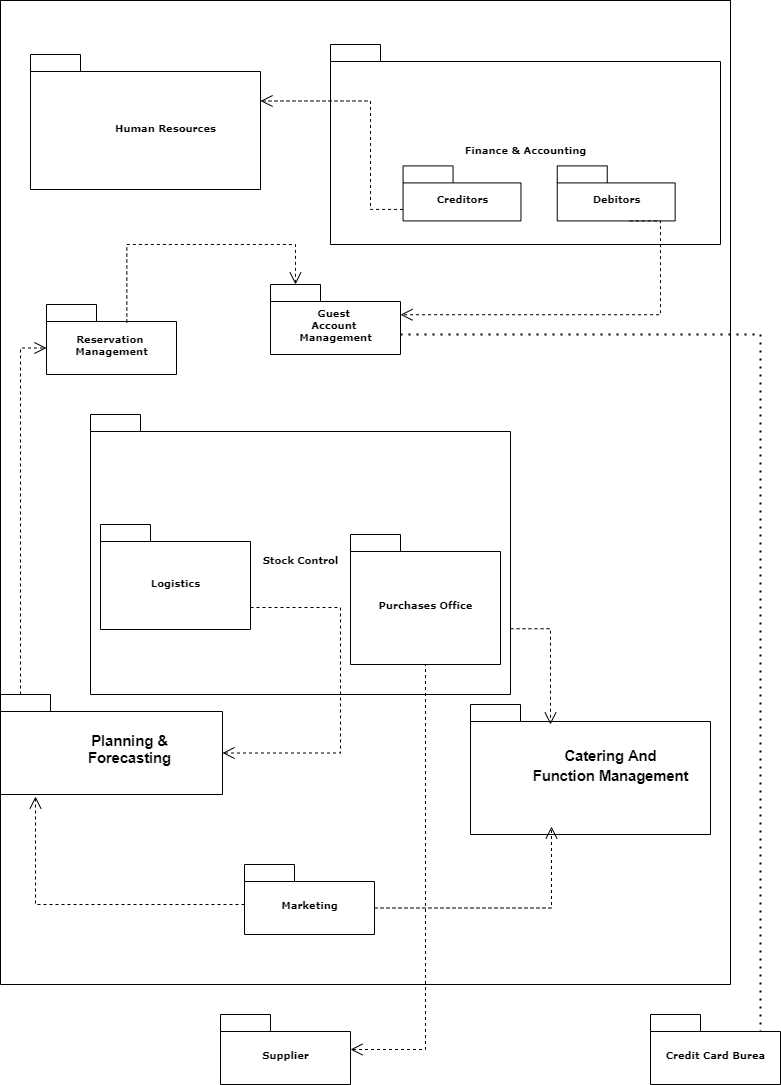
2. The system should allow the receptionist to enquire about guest booking to review its details and determine its status of unconfirmed or confirmed (deposit paid).

3. The system should generate a dynamic electronic report specifying occupancy levels. The receptionist should be allowed to choose the dates for which the occupancy level report will be generated

4. The system should generate any other relevant report which could bring business value to the Phumla Kamnandi Hotel group.

*Guests will phone the hotel to make a booking. The following extract from the “Make a Booking” use case details the required functionality:*

Figure 1.1.1 Package diagram with SUD highlighted displaying the “capture a phone-in” Customer Booking that is to be designed and developed in this project is the Marketing, Guest Account Management, and Reservation Management

**

**Figure 1.1.2 Table showing department and respective responsibilities**

| **Package Name** | **Package Responsibilities** |
| --- | --- |
| **Reservation management** | 1. Manage Bookings 2. Print/ Check Bookings 3. Connect Customer To Hotel Representative 4. Capture Booking Details 5. Capture Customer Details 6. Generate Bookings List |
| **Guest Account Management** | 1. Reservations 2. Cancellations 3. Deposits to Secure Room 4. Check-In 5. Check-Out 6. Room Allocation 7. Room Services 8. Guest Details |
| **Finance & Accounting** | **1.** **Debtors**  · Record Payment  · Check Account Balance  ·  **2.** **Creditors**  3. Maintain General Ledger  4. Maintain Financial Reporting |
| **Stock Control** | 1. Maintain Stock 2. Order Stock 3. Check Stock Levels |
| **Purchasing** | 1. Order Stock |
| **Logistics** | 1. Planning and Replenishment |
| **Human Resources** | 1. Manage Taxation 2. Shift planning 3. Manage leave 4. Recruit new staff 5. Assign training 6. Generate Payroll |
| **Planning & Forecasting** | 1. Manage occupancy |
| **Marketing** | 1. Create And Maintain Catalogue 2. Menu Printing 3. Capture Client Details |
| **Catering and Function Management** | . 1. Manage Activities At Hotel |

The Marketing, Reservation Management, and Guest Account Management packages are the ones that are being designed in this document and developed in the implementation stage of this project because those are the parts that are needed for the requirements outlined above to be met by the system and the Guest booking function. The Reservation Management depends on the Guest Account Management.

Guests will phone the hotel to make a booking. The following extract from the “Make a Booking” use case details the required functionality:

**Figure 1.1.3 Actor table to describe all the actors involved in the system under discussion**

| Actor | System |
| --- | --- |
| ` 1. Customer contacts hotel and asks to make a reservation |  |
| 1. Receptionist enters reservation details into the system | 3. Checks room availability and room price and displays details on screen |
| 1. If accommodation is not available, ask for new dates and return to 2 above. |  |
| 2. Customer confirms the reservation to be made |  |
| 3. Receptionist enters new or existing guest details | 4. Verifies existing guest or adds new guest |
| 5. Receptionist indicates reservation should be created | 6. System creates reservation and guest account. The system calculates the deposit amount and generates a reservation reference numbers. |
| 7. Receptionist enters guest credit card details | 8. System verifies payment with the credit card company. The system records payment and adjusts guest account to record deposit. |
| 9. Receptionist confirms that reservation is complete and advises guest of reservation reference number. | 10. System generates confirmation letters to be emailed, faxed, or posted to guests. |

## Design Assumptions & Constraints

*This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.*

**Design Assumptions**

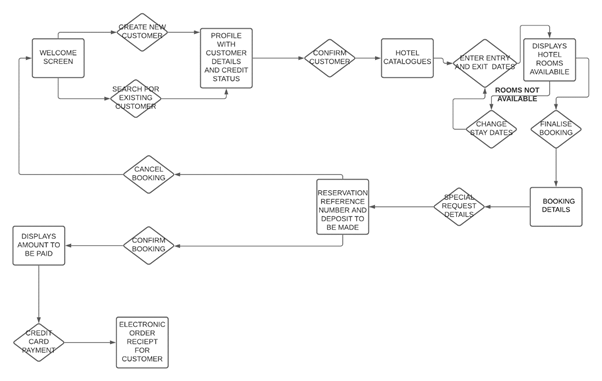
* The system is not required to authorize deposits paid at the time of booking (assume all payments are authorized)

**Design Constraints**

# User Interface & Dialogue Design

## Interface Flow Diagrams

A Flow Diagram is a diagram that depicts the flow of relationships in the system that is being designed, and in the case of the Phumla Kamnandi Hotel, the “phone in” booking system is mapped down below in order to prepare for the implementation stage of this project. The full clear diagram can be viewed  [here](https://drive.google.com/drive/folders/1FUa-SFUQKGSc8M9ZT4MrRDlQLhJHBuMv?usp=sharing)



*Figure 2.1 Flow diagram representing a flow or set of dynamic relationships in the Phumla Kamnandi “phone in” booking system*

## Screen Standards

**For efficient user interaction, the team decided to use the 4 golden rules of design which are as follows:**

1. Place users in control of the interface

• The system navigation techniques were used

2. Make it comfortable to interact with a product

• Minimalistic layout was used with fewer elements so that the user is not confused or intimidated by the system design.

3. Reduce cognitive load

• The system must follow an intuitive design so that even new users don’t have to ask someone to explain to them how the system works. This allows for people even at the beginner level to be able to use the system.

4. Make user interfaces consistent

• Red button to be used for exit or cancel on each page, the green button was used for submit button

• Blue button for navigating to home page

• Text boxes on-page to all be the same length

## Detailed Screen Layout

| **HOME** | The **HOME** screen is the first screen the user will see and an image of the “Phumla Kamnandi Hotel” logo on the left has the following buttons on the right:   1. Telephonic Booking   2. Change Booking  3. Make An Enquiry  4. Company Reports  5. Company Information  6. Close Program    The close program button closes the program |
| --- | --- |
| **TELEPHONIC BOOKING BUTTON PART 1** | When the “Telephonic Booking” button is pressed the following screen pops up and allows the clerk to create a new customer or search an existing one via customer ID.    Inputs required for existing user   1. Customer ID     Inputs required for new user   1. Name 2. Surname 3. Phone Number 4. Home Address 5. Credit Card Number   All these inputs above are varchar and are mandatory  Buttons  1. **Home Button** – Navigates to Home screen |
| **TELEPHONIC BOOKING BUTTON PART 2** | Once a customer is confirmed to exist in the database the following screen pops up with the customer ID at the top of the screen and the following inputs are required    1. Hotel Name  2. Number of Guests  3. Entry Date  4. Exit Date  Then the system displays the total amount for the booking if space is available if not it returns an error message allowing the clerk to enter new details    All these inputs above are varchar and are mandatory.    Buttons  1. **Home Button** – Navigates to Home screen  2. **Submit Button**- Allow clerk to submit a booking for reservation to be made |
| **CHANGE BOOKING BUTTON** | This page pops up when the “Change Booking” button is pressed and allows the clerk to alter the following information about the booking:    1. Name of client  2. Change Address  3. Change Dates  4. Change entry and exit dates  All these inputs above are varchar and are optional.  Buttons  1. **Home Button** – Navigates to Home screen  *2.* **Submit Button**- Allow clerk to submit the changes made |
| **CANCEL BOOKING BUTTON** | This page pops up when the “Cancel Booking” button is pressed and allows the clerk to cancel a booking by using the following:    A. Customer ID  B. Reservation ID    All these inputs above are varchar and are optional  Buttons  1. **Home Button** – Navigates to Home screen  *2.* **Submit Button**- Allow clerk to submit the changes made |
| **MAKE AN ENQUIRY BUTTON** | This page pops up when the “Make An Enquiry” button is pressed and allows the clerk to view the following details about the booking by using the following  1. Customer ID  2. Reservation ID  3. Entry Date  4. Exit Date  5. Room Number  All these inputs above are varchar and are optional.  The text area consists of a list of all reservations that are made.    Buttons  1. **Home Button** – Navigates to Home screen  *2.* **Submit Button**- Allow clerk to submit the changes made and show the respective information |
| **COMPANY INFORMATION BUTTON** | This page pops up when the “Company Information” button is pressed and allows the clerk to view the following details  1. Project information  2. Company Information  Both are displayed in the text area  Buttons  1. **Home Button** – Navigates to Home screen |
| **COMPANY REPORTS BUTTON** | This page pops up when the “Company Reports” button is pressed and allows the clerk to view the company reports and has the following functions:    1. Allows clerk to merge the reports  2. Allows clerk to print the reports  Both are displayed in the text area Buttons  1. **Home Button** – Navigates to Home screen |

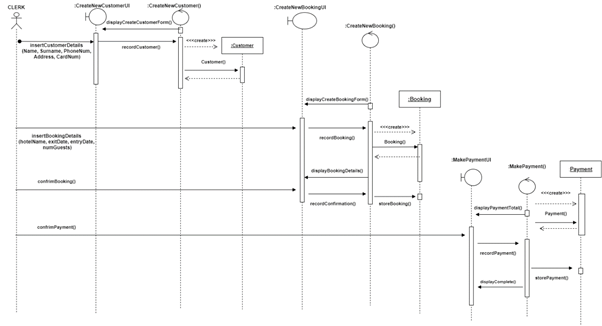
All the wireframes can be viewed [here](https://drive.google.com/drive/folders/1pTBFgSeUZC2UV6s1G31pUPnN05BGZnKC?usp=sharing)

# Design Sequence Diagrams

## Design Sequence Diagram 1

Sequence Design Diagrams are useful for modeling interaction between and system and users in a single-use case, below is a Sequence diagram for the “phone-in” booking use case

*Figure 3.1 Sequence Diagram showing the Phumla Kamnandi “phone-in” booking system interaction with its users (i.e. Clerk), where a new customer is created*



**Assumptions for the Sequence Design Diagram:**

1. There are no bookings that clash with the booking of entry and exit date entered by the clerk.

2. The new customer does not exist in the database hence the system needs to create a new one.

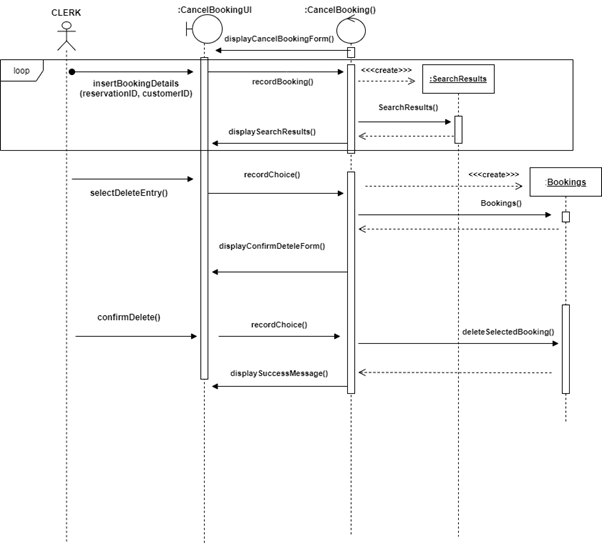
3. Credit status of the customer is approved.

4. Perfect technology.

5. The system displays the final booking information to the clerk instead of printing it.

## Design Sequence Diagram 2

Figure 3.2 Sequence Diagram showing the Phumla Kamnandi “phone-in” booking system interaction with its users (i.e. Clerk), where a customer wants to cancel a booking.



**Assumptions for the Sequence Design Diagram:**

**1.** Clerk needs a reservation ID and customer ID to search for a booking

**2.** The program loops, asking the clerk to input the correct IDs until the customer’s booking appears in the search bar so that the clerk to select it.

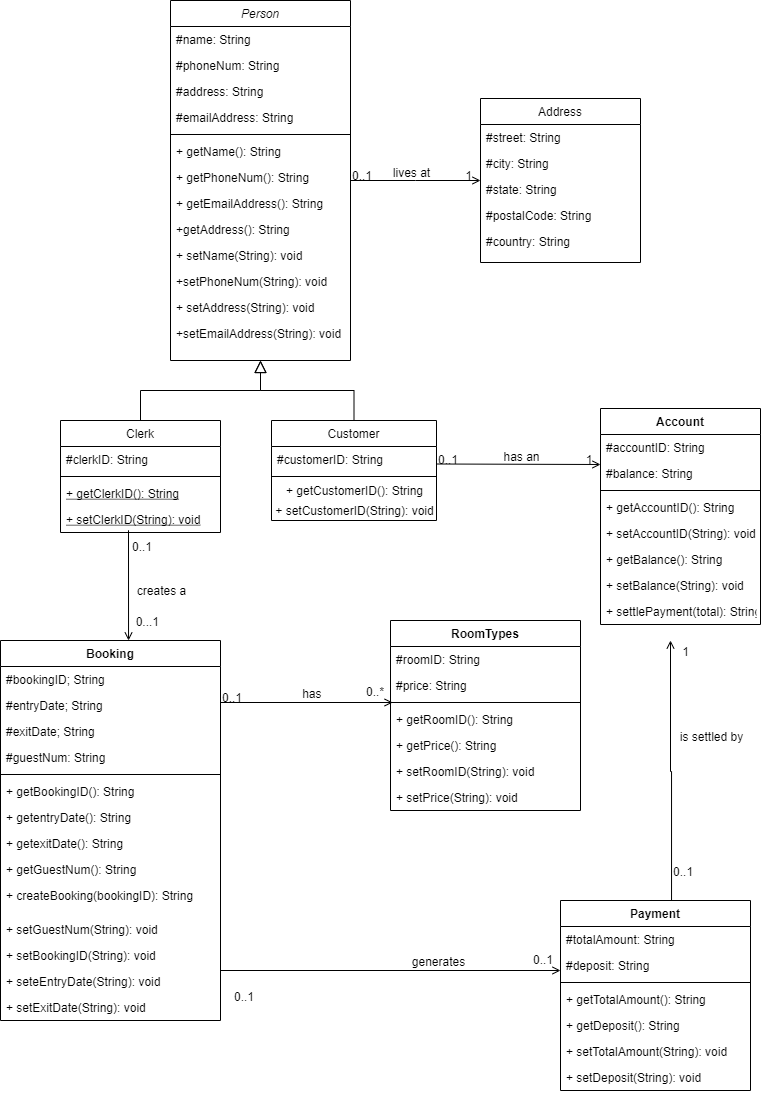
**3.** Program prints out a message stating that the booking has been successfully canceled and automatically updates the reservations list

**4.** Perfect technology

# **Design Class Diagrams**

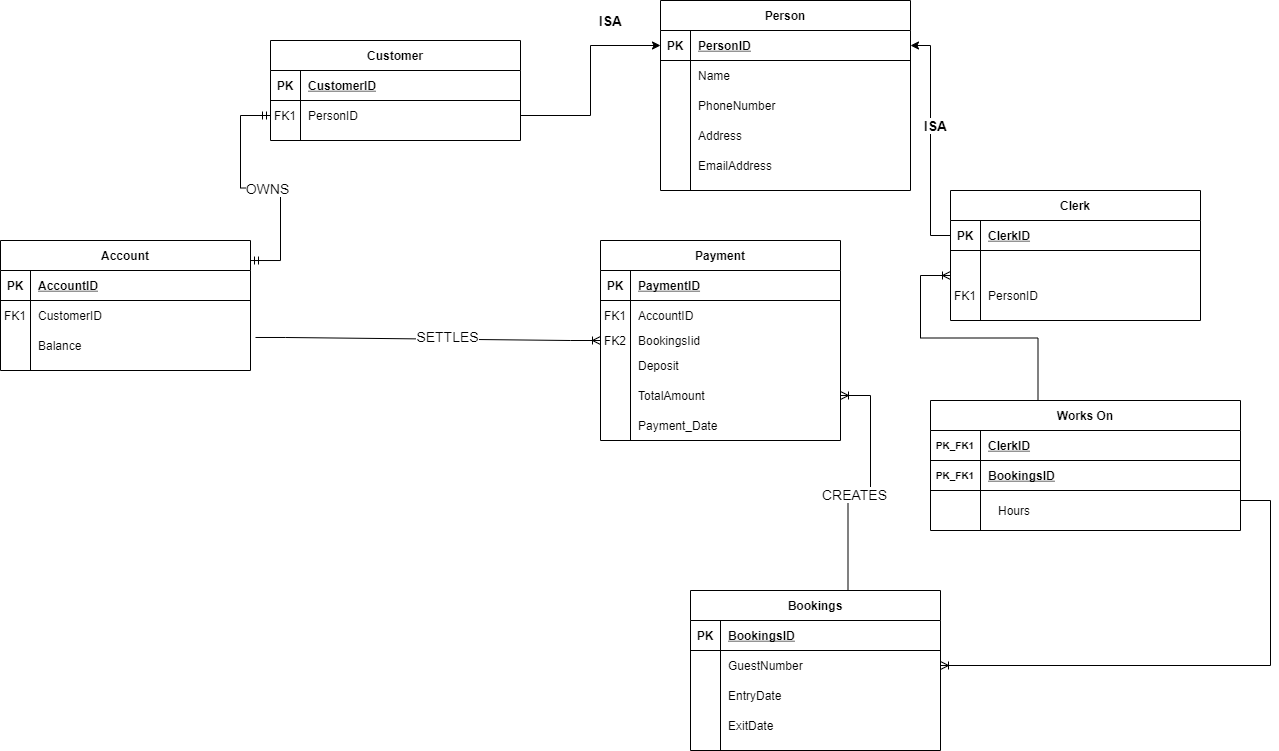
A class diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects, the diagram down below outlines the design structure of the Phumla Kamnandi system.

*Figure 4.1. Design Class Diagram showing the Phumla Kamnandi “phone-in” booking system.*



# **Entity Relationship Diagram**

Figure 5.8 ER Diagram showing the database design for the Phumla Kamndani phone in the booking system

**

**DATA DICTIONARY**

| **Class** | **Description** |
| --- | --- |
| **Customer** | This entity represents customers and contains all customer information for that individual, inherits class “Person”   1. Primary Key: CustomerID int NOT NULL 2. Foreign Key: PersonID int NOT NULL |
| Clerk | This entity represents Clerks, they are responsible for maintaining the customer details and bookings. It has the following attributes:   1. Primary Key: ClerkID int NOT NULL, 2. Foreign Key: PersonID int NOT NULL, |
| Person | This entity is a base class in which the derived Clerk and Customer inherent attributes. It has the following attributes:   1. Primary Key: PersonID int NOT NULL, 2. Name varchar(255) NOT NULL, 3. Address varchar(255) NOT NULL, 4. EmailAddress varchar(255) NOT NULL, 5. Phone Number varchar(255) NOT NULL, |
| Works | This entity represents the interaction between the classes “Clerk” and “Bookings”. It has the following attributes:   1. Foreign Keys: ClerkID int NOT NULL, 2. Foreign Keys: BookingsID int NOT NULL, 3. Hours |
| Account | This entity represents the account owned by the customer and the account that is responsible for settling payment. It has the following attributes:   1. Primary Key: AccountID int NOT NULL, 2. Foreign Keys: CustomerID int NOT NULL, 3. Balance varchar(255) NOT NULL, |
| Booking | This entity represents the account owned by the customer and the account that is responsible for settling payment. It has the following attributes:   1. Primary Key: AccountID int NOT NULL, 2. Entry\_Date date NOT NULL, 3. Exit\_Date date NOT NULL 4. Guest\_Number varchar(255) NOT NULL, |
| Payment | This entity is responsible for recording the payments that the account class makes once the bookings are finalized. It has the following attributes:   1. Primary Key: PayementIDint NOT NULL, 2. Foreign Keys: BookingsID int NOT NULL, 3. Foreign Keys: AccountIDint NOT NULL, 4. Deposit varchar(255) NOT NULL, 5. Total\_Amount varchar(255) NOT NULL, 6. Payment\_Date date NOT NULL, |

# **Report Design**

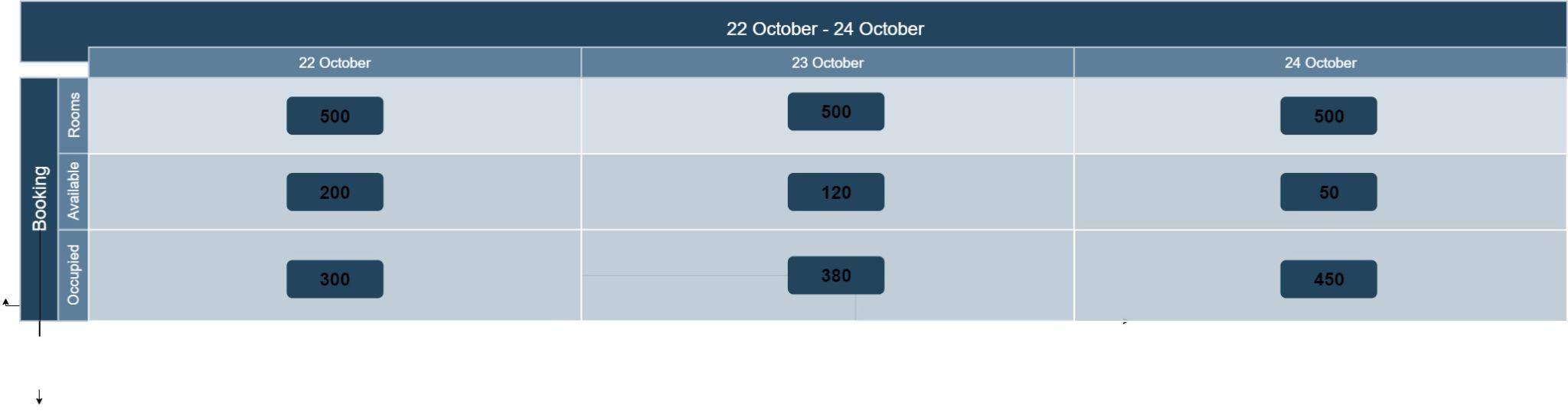
## Operational report

An operational report encapsulates the day-to-day running of the business. This document is very detailed and consists of information pertaining to bookings made by customers.

### Detailed Output Requirements

| ***Output type & ID*** | OccupancyInformation |
| --- | --- |
| ***Report objectives*** | The objective of this type of report is to track, benchmark and improve processes and performance |
| ***Audience*** | Operations Manager and/or stakeholders |
| ***Content*** | Number of rooms booked and available on a specified day or period |
| ***Layout*** | An electronic report that allows users to access the data in real-time |
| ***Selection*** | The rooms that have not been allocated to a customer will be classified as “available” and the once reserved are labelled “occupied”. |
| ***Sequence*** | chronological order of the days |
| ***Comparison*** |  |
| ***Grouping*** | Each day will indicate the number of rooms available and rooms occupied |
| ***Frequency, timing & delivery*** | The information is available in real-time meaning that the information is updated continuously |
| ***Distribution*** | The report will be distributed utilizing a link sent out to the specified audience |
| ***Privacy, security & integrity requirements*** | To minimize the risk of the report leaking, it is sent to the receipt’s professional email. Upon clicking on the link, verification of the individual will be required. It will thus be up to the shareholders to retain the confidentiality of the reports |

### Report Layout



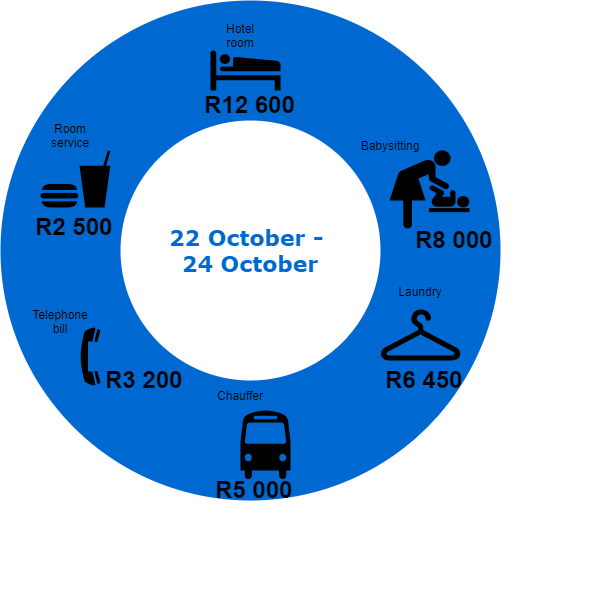
## Detailed report

This report type contains specific information about business transactions.

### Detailed Output Requirements

| ***Output type & ID*** | RoomCosts |
| --- | --- |
| ***Report objectives*** | An elaborate plan that that indicates overall activity, responsibility and resources required |
| ***Audience*** | key members such as the manager |
| ***Content*** | room costs, laundry, room service, babysitting and any additional costs |
| ***Layout*** | *In a graphical manner* |
| ***Selection*** | Hotel services or purchases that are incurred on a guests’ accounts |
| ***Sequence*** | Descending order to the right |
| ***Comparison*** |  |
| ***Grouping*** | The costs incurred by each guest will be grouped into their relevant categories |
| ***Frequency, timing & delivery*** | The data will be evaluated and updated daily by the specified party |
| ***Distribution*** | The document will be emailed to the relevant person/s |
| ***Privacy, security & integrity requirements*** | Only members with authorized access will be allowed to engage with the document. The system of which the document is kept will require authentication |

### Report Layout

**

# **Input-Output Standards & Controls**

*This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator.*

## FFormalized Outputs:

*The formalized forms of output included in the system are the business reports and the Enquiry form*

## Built-In Validation to Ensure Requirements are Met

*The RequiredFieldValidation makes an input control a required field. These required fields include the Customer name and CustomerID*

## Input Integrity Controls

*A data validation control will be implemented to ensure that the details entered in the system are correct and reliable. This control checks information in the tables.*

*A completeness control will also be used to complement the data validation control in ensuring that all-important data such as name, customerID, address and dates are filled in. An error message will appear if the information is omitted.*

## Output Integrity Controls

*Each room should be labelled and identified by a specific number. When customers book a room, a room number should be allocated to the customer on the system to keep track of the number of rooms that are available and which rooms specifically are occupied. The report of occupancy is based on the number of rooms that have been allocated.*

# **Implementation Plan**

*1. Create the Presentation, Business, and Data layer for the system*

*2. Add the entity classes to the Business layer. These would include Customer and Reservation classes as well as the Customer and Reservation Controller classes*

*3. Create a data source*

*4. Create tables and populate them with data*

*5. Connect the system to the data in the SQL database*

*6. Include data layer classes such as CustomerDB and ReservationDB.*

*7. Add boundary classes to the presentation layer. These classes include BusinessReports, CancelBooking, CompanyInformation, HomeScreen, MakeAnEnquiry and TelephonicBooking*

*8. Use test cases to test each use case*

# **Test Plan**

*During testing the software engineering produces a series of test cases that are used to “rip apart” the software they have produced.*

## Test Environment

*The hardware recommended would be a PC or desktop. Minimum software requirements include:*

* *Operating system: Windows 7 or equivalent*
* *CPU: 64 bit support*
* *GPU: GTX 1050*
* *Disk storage: 4GB free space*
* *Monitor resolution: 1280x800*
* *Internet connection*

## Test Items

*The test items are composed of characters, numbers, dates and various options that are selected by a user. CustomerID is a number arranged in chronological order starting at 001. ReservationID is a number arranged in chronological order starting with 1.*

## Test Approaches

*Our chosen methods of testing are unit testing, integration testing, system testing and user acceptance testing.*

* *Unit testing will be used to test individual classes and methods before they are integrated with other software. This process ensures that the separate components perform according to the defined specifications.*
* *Integration testing tests the behaviour of a group of methods and/or classes. This ensures that components are able to perform correctly when executed in combination.*
* *System testing is performed at the end of each iteration. This process tests the whole system to ensure that it meets the functional and nonfunctional requirements.*
* *User acceptance testing determines whether the system fulfills user requirements. The system must satisfy the business needs.*

## Problem Tracking (Test Cases)

| *ID* | *Test scenario* | *Steps to perform* | *Test data* | *Expected results* |
| --- | --- | --- | --- | --- |
| *1.* | *Telephone booking* | *User to log customer’s booking*   * *on the home screen, the receptionist should select “telephonic order”* * *The receptionist then enters the specified details and logs the reservation* | *Customer Name: John Brown*  *CustomerID: 011*  *Street Address: 50 Durban road, Mowbray*  *Suburb: Cape Town*  *Postal code:7700*  *Entry date: 23 October 2021*  *Exit date: 26 October 20201* | *System adds new guest into the Customer table as well as the reservation into the reservation table* |
| *2.* | *Change guest booking* | *Receptionist to change existing booking.*   * *The receptionist selects “change booking” on the home screen. They then enter the required information in the ChangeBooking form* * *the receptionist verifies the change* | *Customer Name: John*  *CustomerID: 011*  *Reservation ID: 11*  *Street Address: 50 Durban road, Mowbray*  *Suburb: Cape Town*  *Postal code:7700*  *Entry date: 26 October 2021*  *Exit date: 30 October 20201* | *The system should change the dates to the updated ones.* |
| *3.* | *Cancel guest booking* | *Receptionist to cancel a current booking*   * *The receptionist selects “cancel booking” on the home screen.* * *This leads them to the CancelBooking form, of which the receptionist should select the “remove reservation option”.* | *ReservationID: 2* | *The booking should be removed from the Customer table* |
| *4.* | *Guest booking inquiry* | *Receptionist inquiries on a booking*   * *On the home screen, the receptionist selects “make an enquiry”* * *They then add the required data* * *Proceed to searching the entry* | *RevervationID: 2* | *A review of the specified booking should be displayed* |
| *5.* | *Business reports* | * *On the home screen, the user selects the “Company Reports”* * *The user enters specific dates* | *dates:  “22 October - 24 October”* | *The system should display the reports on the BusinessReports Form* |

## Test Schedule

| *Test* | *Start date* | *End date* |
| --- | --- | --- |
| *Unit testing* | *20 October* | *21 October* |
| *Integration testing* | *22 October* | *22 October* |
| *System testing* | *23 October* | *23 October* |
| *User acceptance testing* | *24 October* | *24 October* |