*Department of Information Systems*

***Systems Design & Development***

**

**Systems Specification for PKH Booking System**

***Team Members***

|  |  |
| --- | --- |
| ***Student Number: MLNUDZ001*** | ***Student Name: Malindi Udzulanarine*** |
| ***Student Number:*** | ***Student Name :*** |
| ***Student Number: RMVLAS001*** | ***Student Name: Lasy Ramovha*** |
| ***Student Number:*** | ***Student Name :*** |

**Plagiarism Declaration**

1. We know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. This Systems Specification is our own work.
3. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as their own work.

Full Name: \_Malindi Udzulanarine\_\_\_\_\_\_\_\_\_\_ Signed:\_\_\_M.U\_\_\_\_\_\_\_\_ Date: 08\_/10\_/2023

Full Name: \_Lasy Ramovha\_\_\_\_\_\_\_\_\_ Signed:\_\_\_L.R\_\_\_\_\_\_\_\_\_\_\_\_ Date: 08\_\_/\_10\_/2023

**Table Of Contents**

[1. Introduction 1](#_Toc428175304)

[1.1. Overview of Specification 1](#_Toc428175305)

[1.2. Context & Scope of System Specification 1](#_Toc428175306)

[1.3. Design Assumptions & Constraints 1](#_Toc428175307)

[2. User Interface & Dialogue Design 1](#_Toc428175308)

[2.1. Interface Flow Diagrams 2](#_Toc428175309)

[2.2. Screen Standards 2](#_Toc428175310)

[2.3. Detailed Screen Layout 2](#_Toc428175311)

[3. Design Sequence Diagrams 2](#_Toc428175312)

[3.1. Design Sequence Diagram 1 2](#_Toc428175313)

[3.2. Design Sequence Diagram 2 3](#_Toc428175314)

[4. Design Class Diagrams 3](#_Toc428175315)

[5. Entity Relationship Diagram 3](#_Toc428175316)

[6. Report Design 4](#_Toc428175317)

[6.1. Report 1 5](#_Toc428175318)

[6.1.1. Detailed Output Requirements 5](#_Toc428175319)

[6.1.2. Report Layout 5](#_Toc428175320)

[6.2. Report 1 5](#_Toc428175321)

[6.2.1. Detailed Output Requirements 5](#_Toc428175322)

[6.2.2. Report Layout 5](#_Toc428175323)

[7. Input-Output Standards & Controls 6](#_Toc428175324)

[7.1. Formalised Outputs: 6](#_Toc428175325)

[7.2. Built-In Validation to Ensure Requirements are Met 6](#_Toc428175326)

[7.3. Input Integrity Controls 6](#_Toc428175327)

[7.4. Output Integrity Controls 6](#_Toc428175328)

[8. Implementation Plan 6](#_Toc428175329)

[9. Test Plan 6](#_Toc428175330)

[9.1. Test Environment 7](#_Toc428175331)

[9.2. Test Items 7](#_Toc428175332)

[9.3. Test Approaches 7](#_Toc428175333)

[9.4. Problem Tracking (Test Cases) 8](#_Toc428175334)

[9.5. Test Schedule 8](#_Toc428175335)

***Important Notice***

**The comments and suggestions provided in this template must NOT be present in your final submission! Try to customise, apply and re-work this template as much as possible to ensure that your Project’s Value Proposition is maximized** (i.e. Try to avoid just completing this document, heading by heading, and then just updating the contents page). **If nothing else, please just ensure to remove the initial content guidelines provided in this template in each section** (that includes this paragraph!). **The template is only here to assist you in completing your Systems Specification Document, you must try and find ways of showing us that you have actually thought about what each section means, how it creates value, and how it applies to your specific understanding of the problem, and your solution.**

***General Hints/Tips***

1. **The contents page makes use of Microsoft Word’s built in “Table of Contents” feature (found on the far left under the References tab). Try and make use of these helpful built in features** (e.g. once you have completed your document, get Word to automatically ensure that the contents page is accurate and has the correct page numbers by first right-clicking on the table of contents and selecting “Update field” which then updates any altered headings and inserts the correct page numbers)
2. **You should be able to generate an Initial/Draft ERD directly from your SQLExpress database. Doing this is a great way to reduce the amount of work you have to do, while also ensuring completeness. Take the generated model and modify it / comment on it / restructure it for the purposes of your project.**

# Introduction

## Overview of Specification

*Write a short overview for this document (providing a brief overview of the documents structure. Also try to remind your readers of where the project came from (i.e. relevant previous documents and/or work), where the project is currently (i.e. the current completeness of work/documentation, and any resultant changes), and finally what is on the horizon (i.e. what comes next, what work is left to be done, what might require revision, etc)*

This document outlines the detailed specifications and requirements for the information system developed. We use the system specification document to serve as a blueprint for the development of our system. We begin by analyzing our project context and scope and defining our system specification. After the Analysis phase we then move on to the Design phase, this phase is broken down into smaller design phases. This is where we design our User Diagram, Sequence Diagram, Class Diagram and Entity Relation Diagrams including a detailed Design report. The last phase is the Testing phase, we test the Input-Output standards and Integrity Controls to ensure that the requirements are met. We then develop our Implementation plan to move from the design to the complete system as well as a Test plan to rigorously test our system in a series of carefully planned test cases on a controlled environment using carefully planned approach.

## Context & Scope of System Specification

*Write a brief description of the overall project, providing some context and background to the problem, solution, etc. This should quickly remind readers of the project’s relevance and significance, while also reminding them of the objectives and requirements that have been outlined previously, and which are crucial for design.*

*Provide a package diagram to show what particular part of the system is being worked-on/re-designed/developed, ensuring to explain/justify (either explicitly or implicitly) why and how that particular scope/approach was chosen (e.g. dependencies and addressing project critical systems first). Ensure also to provide a valuable scope statement, that will be useful to reaffirm the project’s focus and avoid scope creep.*

The Phumla Kamnandi Hotel group controls the reservation of hotel around South Africa. Phumla Kamnandi wants to improve its reservation system so that it maximizes its income and ensures that its customers are satisfied. This project was created to design the system after the analysis phase has been concluded. The main problems are as follows: -the reservation system that is being used allows booking a room a year prior to the visit. This creates an issue of overbooking the hotel.

-some hotels still use manual registers which may sometimes not be kept up to date, which results in loss of income.

The solution to these problems is to design and create a reservation system that will allow the receptionist to capture the booking and have all the important information about the reservation. This system will ensure that all the business requirements are met.

The system should be able to store the customers' details. I should also capture and store the reservation details for each customer. It should always update the rooms available in each hotel when a reservation is made. These requirements should be met while ensuring that the system is still user-friendly.

## 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.*

The hotel system must integrate with the previous hotel system so that all their previous work and information is not discarded. This can be a constraint for the new design. We may have limited resources to complete the design in a way that we want so this can affect the performance and scalability of the new system. Privacy may be an issue when dealing with people information so the new system may need adjustments depending on how the privacy regulations are.

We assume that the users of the system are going to use the normal email and password log in and they know this information to avoid password resets. We are also assuming that resources like servers and storage will be enough and compatible for the new system. We are assuming that our IT team has the skills to develop the new system without any outsourcing. We also assume that we have enough time to develop the system without causing any delay in business, so the project will be completed on time.

# User Interface & Dialogue Design

*It is a good habit to always write a very short introduction to each new section (i.e. never have heading directly after each other without some explanation. Provide the layout of all input data screens or graphical user interfaces (GUIs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary. This section should contain edit criteria for the data elements, including specific values, range of values, mandatory/optional, alphanumeric values, and length. Also address data entry controls to prevent edit bypassing. Discuss the miscellaneous messages associated with operator inputs, including the following:*

* *Copies of form(s) if the input data are keyed or scanned for data entry from printed forms*
* *Description of any access restrictions or security considerations*
* *Each transaction name, code, and definition, if the system is a transaction-based processing system*

## Interface Flow Diagrams

* *Dialog design can be modelled using a wireframe diagram or Interface Flow Diagrams and the resulting implementation classes can be included in your sequence diagrams.*
* *Ensure to map out and explain the core architecture of the page/form/etc and why you have decided to structure in that particular way.*

*A diagram of a customer service

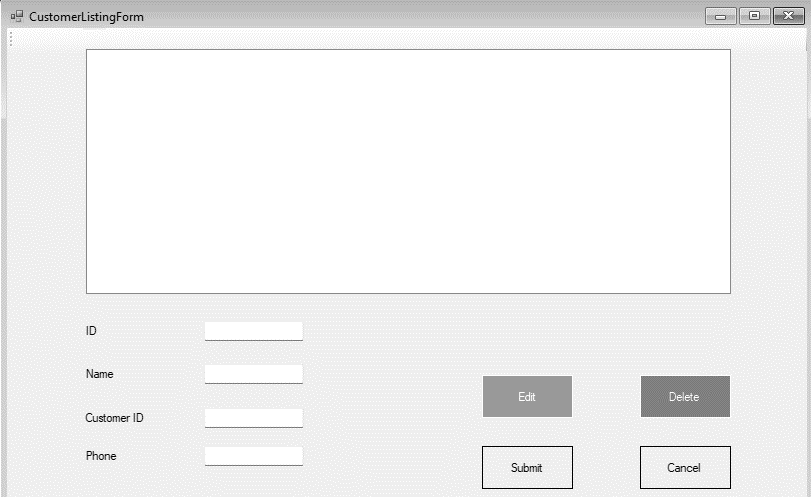
Description automatically generated*

## Screen Standards

* *Once you have identified the various forms and their purpose, define your overall screen standards (layout, colour and overall appearance).*

The layout of the screen is a flexible layout. It changes depending on the screen. The overall appearance is highly functional and visually appealing to make use easier for the user. The colour is a combination of dark and light colour. This is to ensure that everything is visible and attractive to the human eye. Fonts used are very clear and bold enough to read.

## Detailed Screen Layout

* *Design each screen layout in detail.*
* *This can be done by doing “realistic mockups” or by simply adding realistic detail to your wireframe. Use* <https://ninjamock.com/> *to design your screens*.
* **

*A screenshot of a computer screen

Description automatically generated*

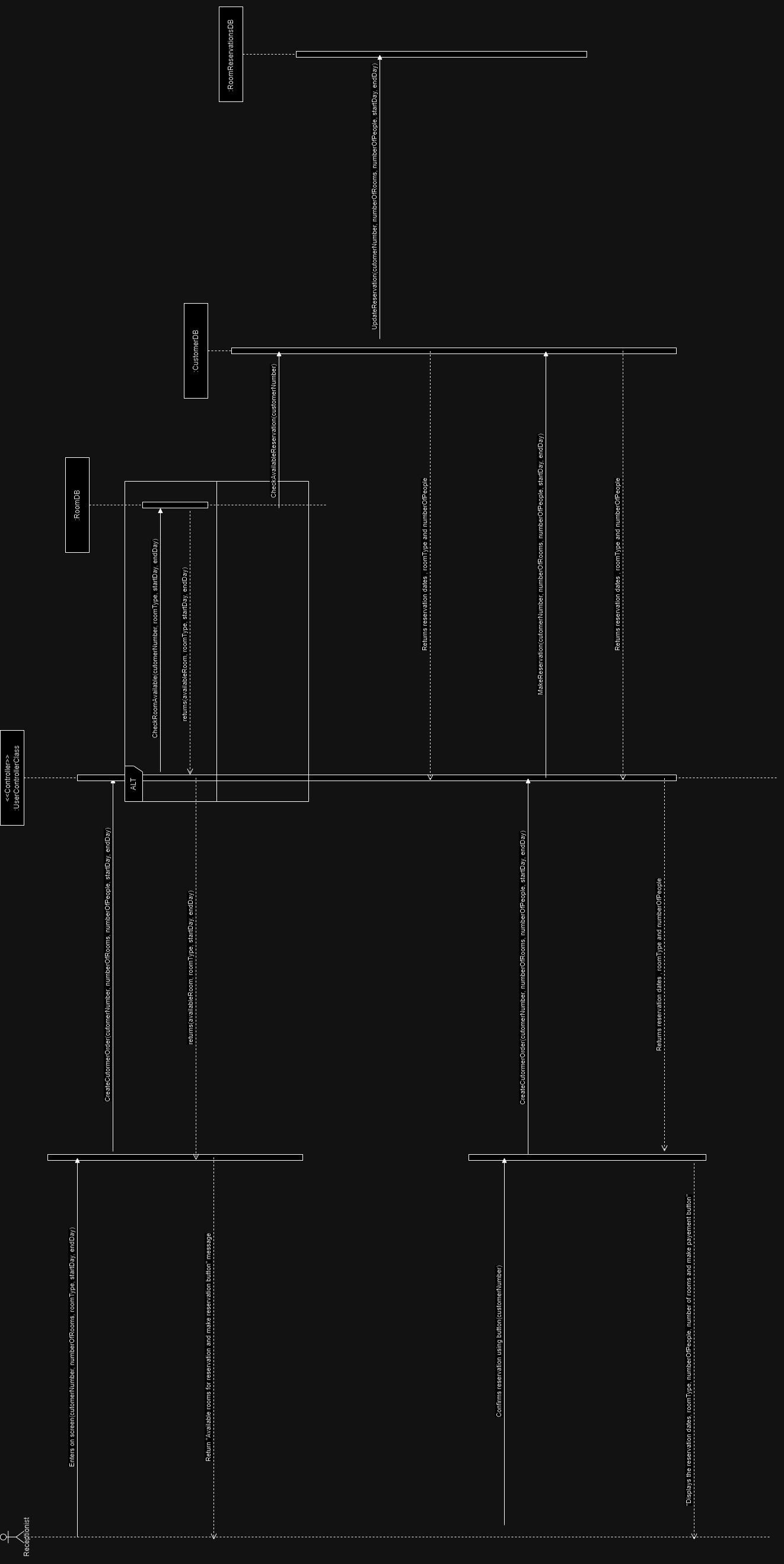
**

# Design Sequence Diagrams

*You are now in a position to model any complex use case interaction in detail using Design Sequence Diagrams, remember the difference between previous sequence diagrams (as utilized during the analysis phase) and the Design Sequence Diagram (which includes, amongst others, possible boundary and controller classes, patterns and detailed messages). Please provide* ***2 design sequence diagrams for two different use cases, one of which should be the “Make a Booking” use case.***

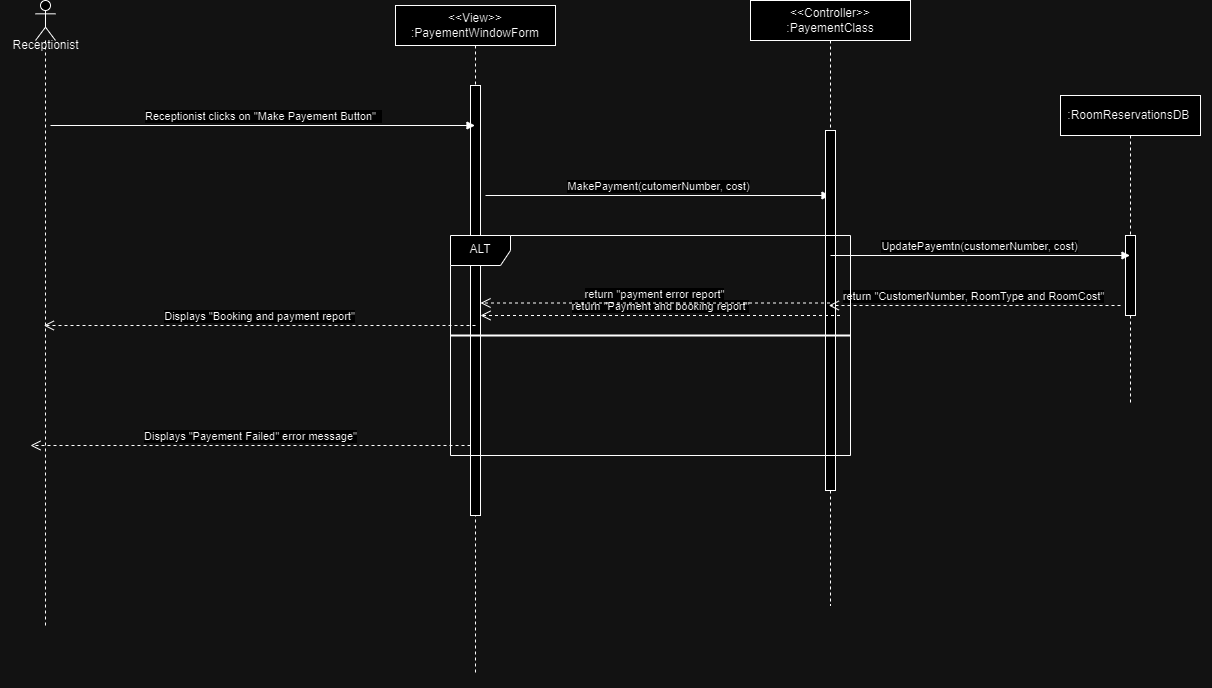
## Design Sequence Diagram 1

* *Ensure to provide some explanation, context and clarification of the model (either in text, or using comments on the model)*



## Design Sequence Diagram 2

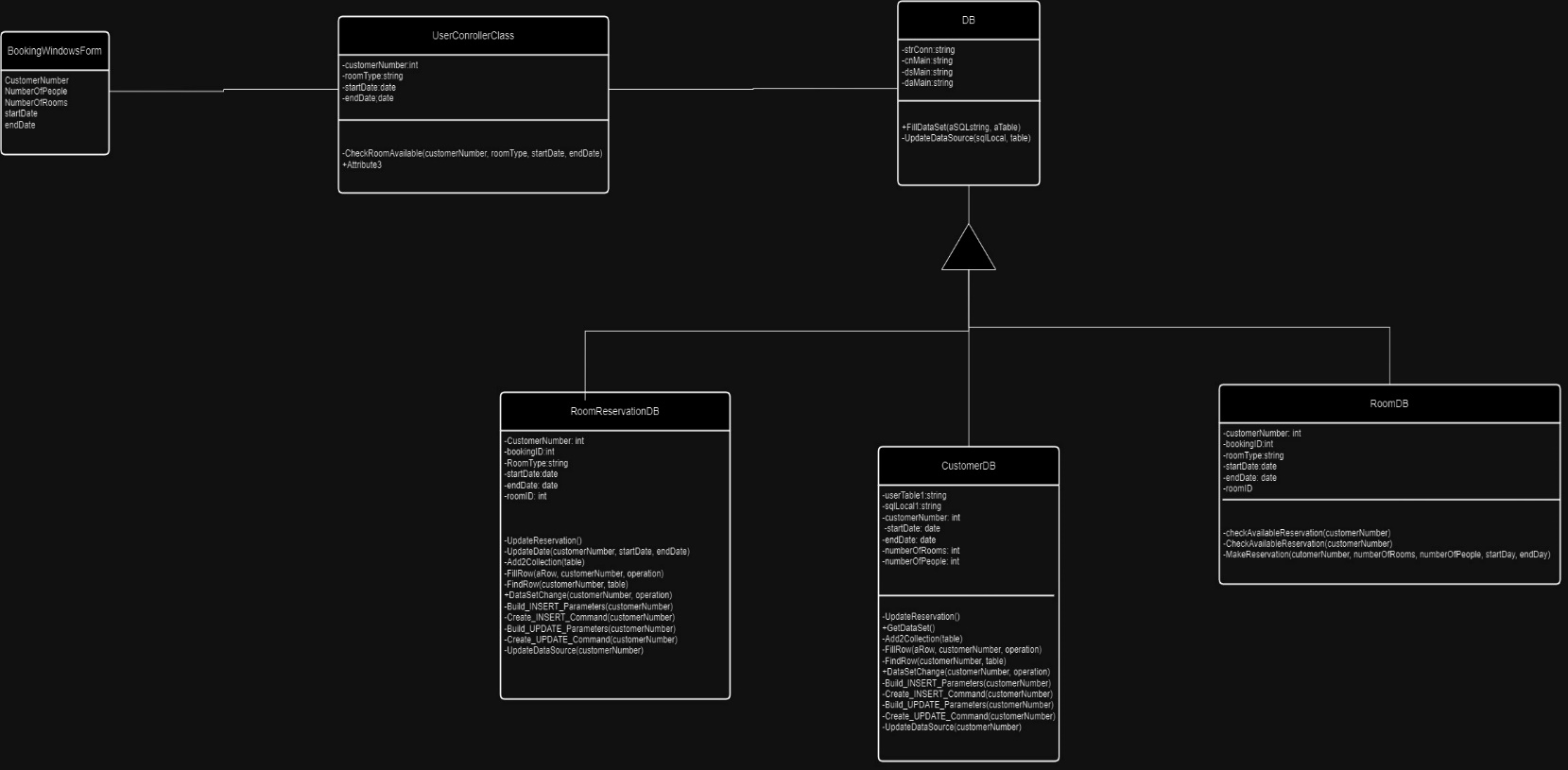
* *Ensure to provide some explanation, context and clarification of the model (either in text, or using comments on the model)*



# **Design Class Diagrams**

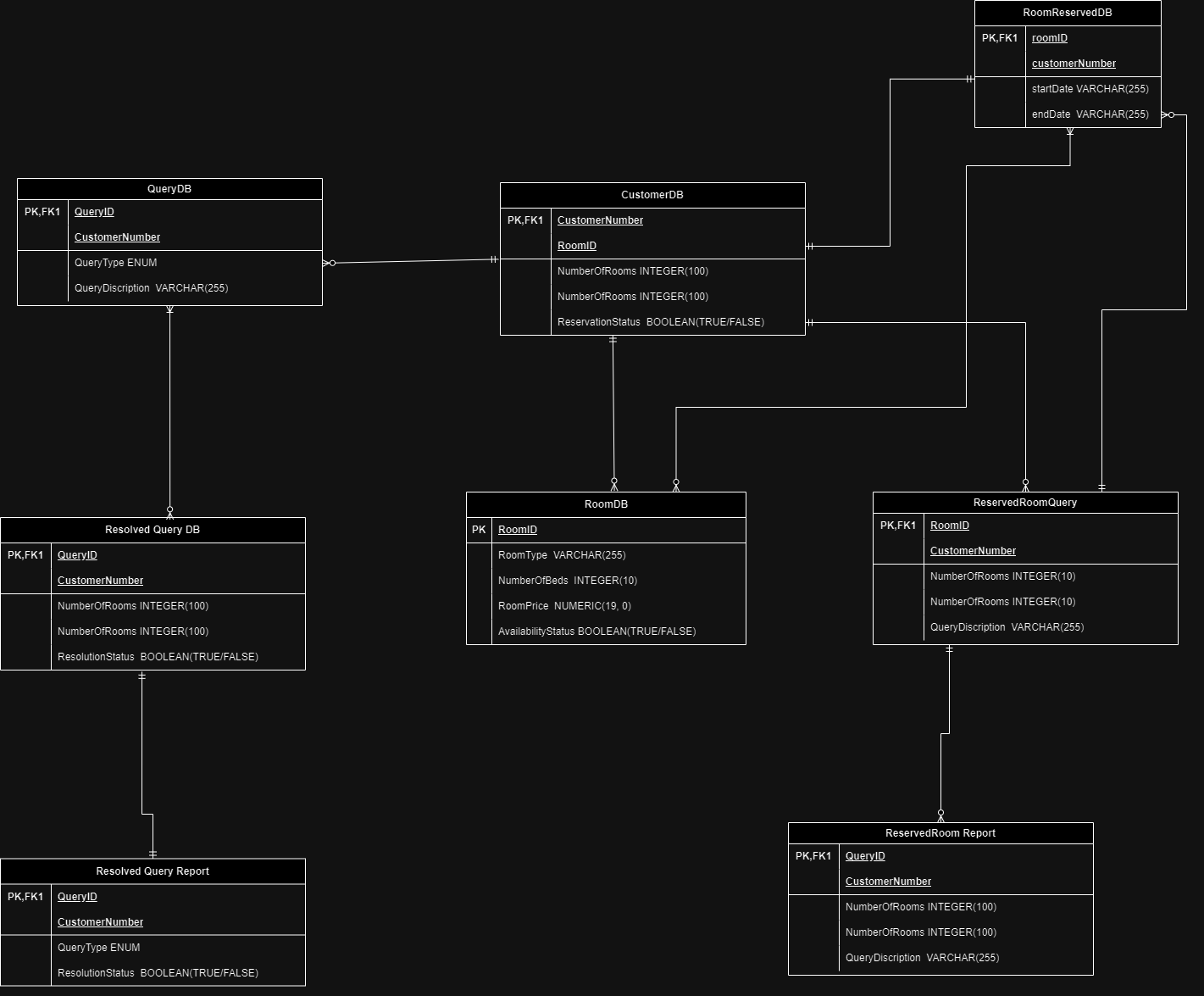
*This is an important set of diagrams building onto your class diagram developed in the analysis stage to describe design components within the classes. You must provide detailed documentation of all classes, their attributes and methods required by the use cases described earlier. Use the standard UML notation for interface classes to indicate where forms are used. (Examples of an expanded class diagram can be seen below)*

*Ensure to provide some explanation, context and clarification of the model (either in text, or using comments on the model)*



# **Entity Relationship Diagram**

*As in most commercial environments today, your system will be implemented using a relational database. Model and Describe your Entity Relationship Diagram (or Database Diagram-DD) using Crows-foot Notation, ensuring all Entities and Associations have correct Cardinality and Multiplicity. Also ensure that your ERD goes beyond simple domain entities, and is fully Normalised to 3rd Normal Form (3NF). Provide detailed documentation on all attributes, keys (primary and foreign), data types and field sizes in a data dictionary table. Include this detail for only the tables required for this iteration of the project.*

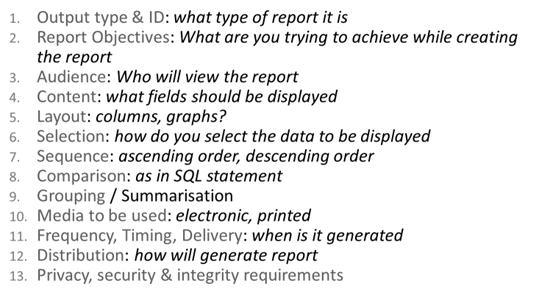
6. Report Design

*This Section will be where you describe the reports that the system will generate. Ensure to add valuable relevant reports only, and to outline complete Requirements Definitions for each Report discussed. You should* ***have 2 reports*** *outlined in this section one of which should be a dynamic electronic report specifying occupancy levels. The receptionist should be allowed to chose the dates for which the occupancy level report will be generated.*

## Report 1

*Briefly describe the report, why it is valuable, the scenario in which it will be generated, who will generate it, why, etc..*

### Detailed Output Requirements

******

1. Summary report 1001
2. Report Objective: Give a summary of the periodic activity of staying in the hotel.
3. Audience: The report will be view by the receptionist and the customer
4. Content: Customer details and reservation details
5. Layout: Column form
6. Selection: The information is generated depending on the option selected using different buttons.
7. Sequence: Ascending order is used
8. Comparison
9. Grouping: Grouped customer
10. Media: Electronic copy is sent to the user
11. Frequency: The report is generated whenever requested
12. Distribution: The report is generated by the receptionist
13. Privacy: The receptionist ensures that the report is given to the right person and can be the only one who accesses the information.

### Report Layout



## Report 1

### Detailed Output Requirements

1. Detailed report 1002

2. Report Objective: Reports specific information on the business transactions

1. Audience: The report will be for the receptionist and the stakeholders
2. Content: Transaction and reservation details
3. Layout: Column form and graphs
4. Selection: The information is generated depending on the option selected using different buttons.
5. Sequence: No specific order required
6. Comparison
7. Grouping: information is grouped per hotel
8. Media: Electronic copy is sent to the user
9. Frequency: The report is generated whenever requested
10. Distribution: The report is generated by the receptionist
11. Privacy: The receptionist ensures that the report is given to the right person and can be the only one who accesses the information.

### Report Layout

-Tile of the report

-Table of content

-Executive summary

-Introduction

-Overview and key hotel information

-Performance

-Financial reports

-Customer and room statistics

-Maintenance

-Conclusion and appendices

# **Input-Output Standards & Controls**

*This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs*

*Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:*

* *security to restrict access of critical data items to only those access types required by users*
* *Application audit trails to dynamically audit retrieval access to designated critical data*
* *Standard Tables to be used or requested for validating data fields*
* *Verification processes for additions, deletions, or updates of critical data*

## Formalised Outputs:

*All key system outputs should be displayed on the screen. You should be able to display the confirmation letter on the screen if requested, but no other forms of output (printed report, email and fax) are required. What were the formalized forms of output included in your system?*

The system can produce error messages to the user whenever the user input does not correspond to what the system is looking for. This will allow the user to make changes and input the correct information. Once the booking or reservation is completed, a confirmation message is sent to the screen to indicate the successful reservation.

## Built-In Validation to Ensure Requirements are Met

*What were the validation controls built into the system to optimize the performance of the application and to minimize the possibility of user error which may hinder system performance?*

The system has user interface validation which prevents the user from selecting and storing information that is incorrect. For example, booking a hotel from 12 November 2023 to 08 October 2023. The system provides feedback immediately allowing the receptionist to make changes. The error message provided as feedback gives clear instructions on what to do and what to change. The system also shows if the selected reservation dates are fully booked to prevent overbooking the hotels.

Access control and authorization has also been implemented. This will allow only the right people to access the data to prevent data inconsistency. Doing it this way will prevent the loss of income and increase customer satisfaction.

## Input Integrity Controls

*Describe any input integrity control included in the system*

The system has SQL injection prevention. This prevents any manipulation of the data by using parameterized queries. The details being added into the system are given to the user for confirmation. The data type of the input is also verified to make use it is valid and prevent any errors.

## Output Integrity Controls

*Describe any output integrity control included in the system*

The output that is displayed cannot be edited to ensure that the data is not changes and it stays secure. Outputs like confirmation is not presented if there was an error in the process to prevent storing incorrect data.

# **Implementation Plan**

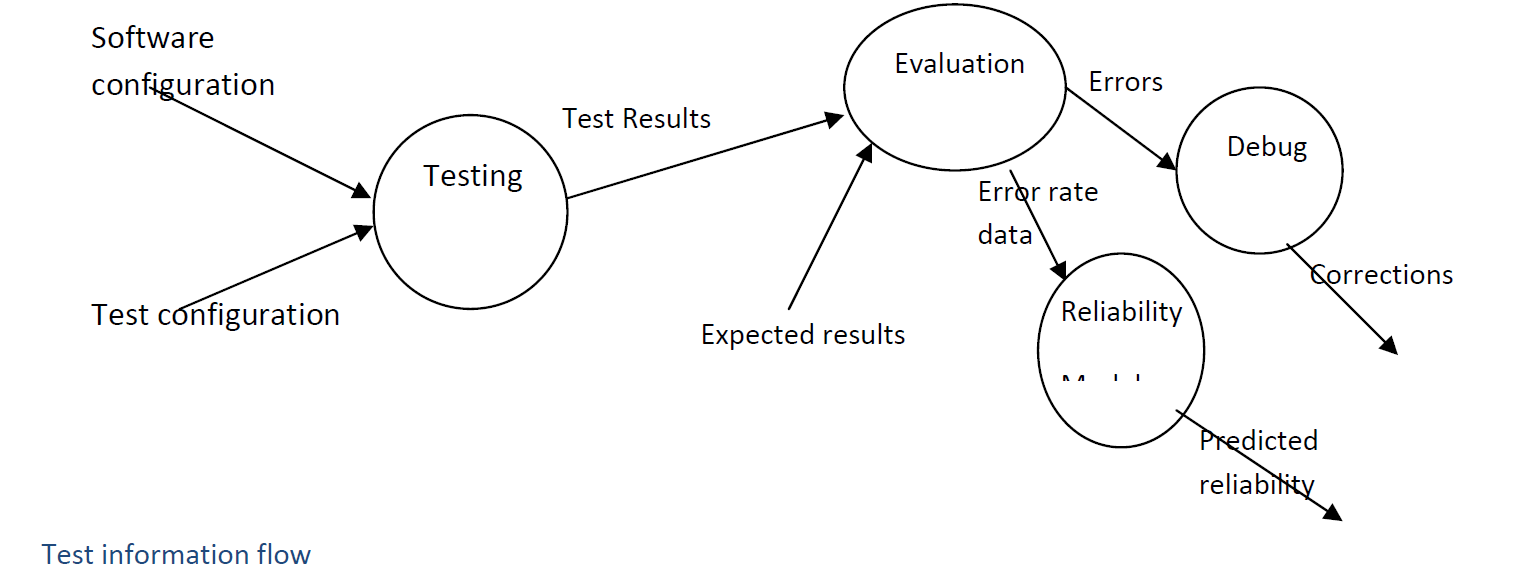
*Include a detailed implementation plan scheduling the tasks your team must complete to move from the design stage to the final delivery of this phase of the system to the users.*

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | Completion period | Designation |
| Review requirements | All the requirements need be revisited and understood so that developer will know what the must achieve. | 1 week | Managers |
| Write and develop the code for the system. | Programmers need to write code for the system and developer must develop the whole system | 8 weeks | Programmer  Developer  Software engineer |
| Testing and checking funtionality | The system needs to be tested and we need to make sure the expected functionality is met. The tests are done using different test cases to make sure that it works well | 7 weeks | Developers |
| Docummentation | Everything about the project needs to be documented so that it is in writing for future use | 1 week | Administrator |
| Training | The users need to understand the new system and know how it works. They should be able to use and should not have any issues, so they have to be trained. | 2 weeks | Project manager |
| Deployment | After the everything has been done, the system can be deployed, and the users can stare using it. | 1 week | Project manager |

# **Test Plan**

*During testing the software engineering produces a series of test cases that are used to “rip apart” the software they have produced. Testing is the one step in the software process that can be seen by the developer as destructive instead of constructive. Software engineers are typically constructive people and testing requires them to overcome preconceived concepts of correctness and deal with conflicts when errors are identified. A number of rules that act as testing objectives are:*

* *Testing is a process of executing a program with the aim of finding errors.*
* *A good test case will have a good chance of finding an undiscovered error.*
* *A successful test case uncovers a new error.*



## Test Environment

*Describe the minimum hardware & software requirements*

The minimal hardware and software requirements are crucial to the efficient and dependable operation of a web-based booking system. A server with a dual-core processor, 4 GB of RAM, plenty of storage, and a quick network connection are requirements for hardware. Your arrangement may call for distinct servers for the web and database components. Additionally crucial are environmental factors like cooling and power backup.

The selection of an operating system, web server software, and a database management system are all required for software. In addition, the project requirements should be met by the programming languages and frameworks, monitoring tools, backup solutions, version control, security precautions, and email services.

## Test Items

*Provide a description of all the features to be tested*

Functionality: The business requirements should be met. Things like log ins, reservations, cancellations, creating customer profiles, editing bookings, etc should be done properly by the system without errors.

Security: Information about customers should be protected and not everyone should have access to the information. Account details should or banking details are an example of things that need to be protected and not displayed unless requested by the correct individuals.

Data integrity: The system should not just allow any changes to the data in the database so that it stays accurate and consistent.

Integration: The system’s integration ability should be tested to all other processes to work as planned. An example of this is the payment process that will allow the customers to secure their reservations.

The user interface: The UI should be tested to ensure that it is user-friendly and all the functionality behind it will work properly after the interaction with the user.

Performance: The system’s performance is very important because if it does all the requirements but has very bad performance, the users will be unhappy, and it will decrease profit for the business.

Usability: The user experience should be tested, and changes should be made to make it more favorable by the users.

## Test Approaches

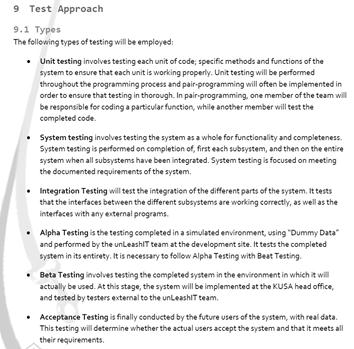
*Describe the types of tests which will be performed*

System testing

The whole system will be tested to make sure that all the business requirements are met. This will prevent any unexpected errors when the system is implemented. All the dugs from all the subsystems will be detected and fixed. Security measures are tested, and we ensure that they are working keep the data secure as planned.

Alpha testing

An alpha test will be performed using test data that will be like what the system will be getting as input after it has been implemented. This will allow us to see if the expected actions and output is being produces. This is also where functionality is tested when we have data that we are using. This will give us the confidence that the system will function as expected.

**

## Problem Tracking (Test Cases)

*Describe the process followed for tracking and resolving errors*

During execution of the test cases, errors will be identified by comparing the output produced with the expected output. Any errors will be fixed until the expected output is produced. Information in the data based is also view and check for any errors that were not detected when entering the data. This information is edited, and the mistake is fixed. These errors are fixed by editing code and retesting the code.

## Test Schedule

*Describe how the test schedule was co-ordinated with the development schedule*

The test schedule and the development timetable must be linked for a successful and effective software development process. Project planning is the first step, during which time milestones and deadlines are set for both the development and testing stages. Testing and development typically go hand in hand in iterative or agile techniques, with developers concentrating on the current sprint and testers getting ready for the upcoming features. To tackle challenges and concerns, close collaboration with frequent meetings and communication are essential. When necessary, test environments must be available, and resource allocation and defect management should be coordinated. The coordination of user acceptability testing, regression testing, and the final release ensures that the product is thoroughly tested and follows the development roadmap. This coordination promotes working together, shortens delays, and eventually results in quick delivery of excellent software.

