ORCHID INTERNATIONAL College

SUPERVISOR’S RECOMMENDATION

I hereby recommend that the report prepared under my supervision by Nitisha Timalsina (TU Exam Roll No. 23879/076), Subrat Regmi (TU Exam Roll No. 23894/076), Supriya Shree Basnyat (TU Exam Roll No. 23900/076) entitled “**HOTEL BOOKING CANCELLATION PREDICTION SYSTEM**” in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

…………………..…….

Er. Dhiraj Kumar Jha

Project Coordinator,

Department of CSIT Orchid International College

Bijayachowk, Gaushala

ORCHID
INTERNATIONAL College

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by Nitisha Timalsina, Subrat Regmi and Supriya Shree Basnyat entitled "Hotel Booking Cancellation Prediction System” in partial fulfilment of the requirements for the degree of B. Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………… **Er. Dhiraj Kumar Jha** Supervisor, Orchid International College Bijayachowk, Gaushala	…………………… **Er. Dhiraj Kumar Jha** Head of Deprtment, Orchid International College Bijayachowk, Gaushala
…………………… **Internal Examiner** Orchid International College Bijayachowk, Gaushala	…………………… **External Examiner** Central Department of Computer Science and IT, Tribhuvan University, Kirtipur, Nepal

ACKNOWLEDGEMENT

The team would like to express our sincere gratitude to everyone who has assisted us on our journey to completion of this project. We are deeply thankful to Er. Dhiraj Kumar Jha, our project supervisor, for his invaluable guidance, support, and encouragement throughout the course of this project. His exceptional expertise and mentorship have been immensely helpful to us in overcoming many challenges.

We would also like to extend our appreciation to the Department of Computer Science and Information Technology, Orchid International College for providing the necessary resources and a helpful environment for developing this project.

Lastly, we would like to express our thanks to the teachers, mentors and our colleagues for their support, understanding, and encouragement throughout this endeavour.

Sincerely,

Nitisha Timalsina,

Subrat Regmi,

Supriya Shree Basnyat

ABSTRACT

Booking cancellations have a substantial influence on the accuracy of demand prediction in the hotel industry. Hotels frequently implement tight cancellation policies and overbooking tactics in order to avoid losses, limiting the number of bookings and revenue. As a result, in the hotel sector, a detailed prediction is a key tool. Models for forecasting a booking's cancellation can be developed to overcome the ambiguity generated by cancellations. Hotel Booking Cancellation Prediction System is a proposed project that aims to build a machine learning model that uses decision tree and random forest algorithm to anticipate hotel room cancellations in order to meet customer expectations and enhance revenue management. The hyper parameter tuned model used in the project provided 81.84% balanced accuracy utilising 35019 instances of data. The proposed system helps in revenue management, optimal resource allocation, reduced overbooking conditions which is beneficial to the hospitality sector.

Keywords: ***Booking Cancellation, Decision Tree, Random Forest, Hyper parameter, Balanced Accuracy***

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
AUC	Area Under the Curve
CASE	Computer Aided Software Engineering
CSS	Cascading Styling Sheet
EDA	Exploratory Data Analysis
GBM	Gradient Boosting Machine
HTML	Hypertext Markup Language
ID3	Iterative Dichotomiser3
IDE	Integrated Development Environment
KNN	K-Nearest Neighbors
MLP	Multi-Layer Perceptron
MVC	Model View Controller
MVT	Model View Template
PNR	Personal Name Records
RDBMS	Relational Database Management System
SQL	Structured Query Language
SVM	Support Vector Machine
UML	Unified Modeling Language

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the dynamic and fast-paced hospitality sector, successfully managing hotel reservations is critical for hoteliers. The uncertainty connected with booking cancellations is one of the issues that hotels encounter.

Hotel Booking Cancellation Prediction System is a web-based application that helps in mitigating the risks associated with reservations using machine learning algorithms. This web system provides two user bases, customers and employees. Customers can make their reservations for hotel rooms through the app while employees can view the reservations and make predictions on whether the customer is likely to cancel or not. This system attempts to give hoteliers useful information to improve decision-making, and optimise revenue management by utilising tree-based machine learning algorithms.

1.2 Problem Statement

Hoteliers in the hospitality sector have the difficult challenge of efficiently handling reservations in the context of unpredictable cancellations. These cancellations have a real influence on revenue management and customer satisfaction in addition to interfering with operational planning. One major obstacle that leads to operational inefficiencies and inefficient use of resources is the lack of a systematic method for forecasting reservation cancellations. Moreover, a flexible solution is required due to the dynamic nature of the hospitality industry, which is impacted by various factors.

The necessity for an extensive booking cancellation prediction is highlighted by the fact that many hotels continue to use outdated approaches and lack the resources to fully utilise data analytics and machine learning for predictive insights into booking cancellations. Machine learning plays a crucial role in predicting hotel booking cancellations by analysing patterns, trends, and various factors that contribute to customer decisions.

In conclusion, the development and implementation of the Hotel Booking Cancellation Prediction System addresses the complex challenges faced by the hospitality industry, providing a strategic and proactive solution for effectively managing and mitigating the impact of reservation cancellations.

1.3 Objectives

The primary objectives of the project are:

- To implement a random forest algorithm from scratch and several optimization strategies for model development.
- To use machine learning for forecasting cancellations of reservations.
- To create a web application and incorporate a model that reliably classifies reservations.

1.4 Scope and Limitations

The system takes data from customer’s booking requests such as meal plan, number of adults, number of children, average price per room, room type, lead time, etc. as inputs to the machine learning model which then predicts the cancellation of bookings based on these provided data. The model uses a random forest algorithm to predict booking cancellations.

The limitations of the system are as follows:

- This system considers a limited hotel booking dataset.
- This system doesn’t provide the count of available rooms.

1.5 Development Methodology

Plan-driven incremental development approach is used for developing this project. Incremental development model helps in building software systems components-by-components in which the final requirements are clear from the start. It involves developing the initial version, providing feedback and changing software through different versions until the required system is developed. Hence, the incremental methodology develops the system as a series of increments or versions, with each version adding functionality to the previous version.

The following figure shows the different phases of incremental development:

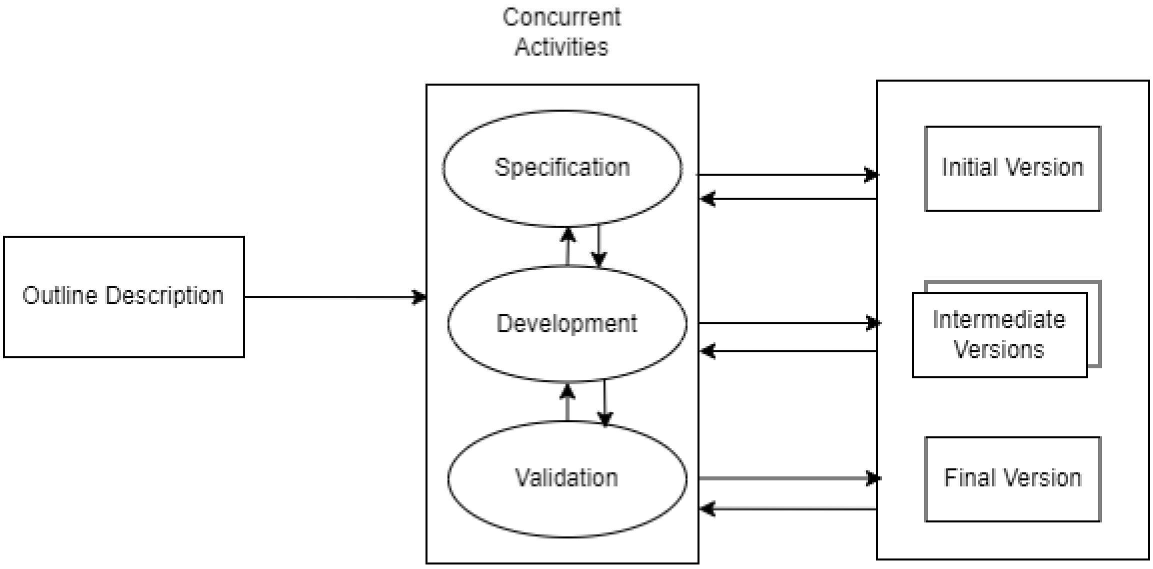


Figure 1.1: Incremental Model

1.6 Report Organization

The report consists of six chapters which are organised as follows:

Chapter 1: Introduction - This chapter includes a thorough project introduction, problem statements that correspond with the project, the project's objectives, its scope and limitations, and the methods employed to build the system.

Chapter 2: Background Study and Literature Review - This chapter offers a review of earlier research on the issue. It covers various study related to data processing and machine learning techniques in order to implement the project.

Chapter 3: System Analysis - This chapter explores both the system's functional and non-functional requirements. It also includes a feasibility study conducted to examine the system's operation and the system's work breakdown structure.

Chapter 4: System Design - This chapter includes detailed design of the system along with algorithm details.

Chapter 5: Implementation and Testing - This chapter covers the hardware tools, dependencies, and software tools needed to complete the project. Several project implementation steps are outlined in this chapter along with various test cases.

Chapter 6: Conclusion and Future Recommendations - This chapter analyses the outcome of the model that was put into practice. It also offers additional tasks that can be performed to improve the project.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Classification Algorithms like the Random Forest Classifier are designed to assign predefined labels to input data based on patterns observed during the training phase. A classifier's purpose is to learn a mapping between input features and the appropriate output labels in order to generate accurate predictions on previously unseen data. Random Forest Classifier is an ensemble learning approach based on decision trees. Random Forest divides the training data into multiple subsets, with each subset utilised to train a decision tree. The number of people, arrival date, special services, car parking, and other factors all play a role in the outcome of a hotel reservation. The algorithm divides the outcome into two categories: likely to cancel and unlikely to cancel. Multiple performance indicators like Accuracy, Precision, Recall, F1-Score, and the AUC-ROC Curve will be examined to test the model's performance.

2.2 Literature Review

Cancellations are a crucial component of hotel revenue management due to their effect on reservations for rooms but only little is understood by hoteliers about why consumers cancel their bookings or how to prevent it. Because of the effect that cancellations have on hotel chains, hotels come up with various tactics for the express purpose of minimising cancellations, which in turn has an influence on hotel income and reputation. These factors make it essential for hotel management to get early notice of cancellations. Demand forecasting and revenue management have a significant correlation and this hole may be filled by utilising machine learning and various algorithms to identify those who are likely to cancel.

In order to develop domain expertise for the project, current research papers that have been published in the specialised fields of machine learning and hospitality were thoroughly analysed. These publications' insights, which aided in supplying knowledge about the field, are mentioned below.

A Study by Eleazar C. Sanchez, Agustin J. Sanchez-Medina, Monica Pellejero on ‘Identifying Critical Hotel Cancellations using Artificial Intelligence’ published in 2020, conducted a comprehensive analysis on the most important factors that affect hotel booking cancellations. The research focused on predicting hotel booking cancellations made close to the time of service. The main purpose of this research was to help hoteliers improve their strategies for maximising revenue while minimising risks and losses. Multiple Artificial Intelligence and Machine Learning models were applied to Personal Name Records (PNR) data which produced great results. The dataset was provided by a 4-star hotel located in Gran Canaria, Spain containing more than 10,000 booking records between 2016 and 2018. A total of 13 columns were used as independent variables and the state of booking (cancelled or not cancelled) was used as the dependent variable. The independent variables included nationality, number of nights, hotel type, previous cancellations, number of adults, number of children, and more. Models were developed on R statistical software using the following packages: C5.0, Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Gradient Boosting Machine (GBM), among which GBM yielded the best results with a 73% accuracy in the worst case and an 80.5% accuracy in the best case. On the other hand, ANN produced the least favourable output with an accuracy of 60% and 69% in the best and the worst cases respectively [1].

Another research conducted by Yaqi Lin in July 2023, dealt with identifying the factors that have the greatest impact on hotel booking cancellations through Exploratory Data Analysis (EDA) visualisation. Due to external uncertainties such as flight cancellations, itinerary changes, conference cancellations, many customers choose to cancel their orders after booking a room [2]. Machine Learning algorithms were applied to guess whether a customer will cancel their bookings. It was found that city hotels have a cancellation rate that exceeds that of resorts by about 12%. Moreover, July and August are the peak period for customer orders, and it is also the maximum number of cancelled orders. Other factors, such as market segment, payment deposit, advanced booking were also found to have major impacts on booking cancellations. Among the three algorithms used, logistic regression was the weakest and random forest showed the highest accuracy results.

A paper by Rachel Mytnik, published on June 25, 2021 aimed to find the best classification model for predicting hotel booking cancellations and find the most relevant explaining variables for customer cancellations [3]. This research paints a clearer picture on hotel booking demands. Several classification models were used in this research, including XGBoost, Random Forest, Linear SVM, and Logistic Regression amongst which XGBoost showed the highest level of accuracy of 99.7%. Linear SVM showed the lowest level of accuracy of 78.65% which was attributed to its linearity. Lastly, the variables that had the highest impact on the model were found to be deposit type followed by required parking space, previous cancellations, and so on.

The popularity of online reviews is causing a huge impact on consumers' purchase intentions for goods and services [4]. However, and hidden by the anonymity of the Internet, fraudsters can try to manipulate other consumers by posting fake reviews. Research carried out by M.R. Martinez-Torres, and S.L. Toral used a content analysis approach based on a set of unique attributes and sentiment orientation of reviews to identify deceptive reviews in the hospitality sector. The main contributions of the paper are a set of polarity-oriented unique attributes able to distinguish deceptive and non-deceptive reviews and the main topics associated with deceptive and non-deceptive reviews. The training data set was processed using various machine learning algorithms including KNN, Logistic Regression, SVM , Random Forest, Gradient Boosting, and MLP. Analysing the data, it was found that deceptive positive reviews often emphasise the location of the hotel while positive non-deceptive reviews are focused on the characteristics of the city. The topics of negative deceptive opinions were focused on complaints about the hotel environment, room environment, etc. and in the case of non-deceptive negative opinions, the complaints are related to long waits, staff behaviour, smell, etc.

Cancellation prediction models are advantageous because they classify the cancellation outcome of each booking and allow an understanding of how each feature influences cancellations, that is, an understanding of cancellation drivers [5]. This study on 'Big data in Hotel Revenue Management' by Nuno, Almeida and Luis determined that by identifying the features that are most important in predicting the outcome of a booking, the cancellation drivers can be narrowed down.

CHAPTER 3: SYSTEM ANALYSIS

3.1 System Analysis

3.1.1 Requirement Analysis

An essential stage in the creation of any system is requirement analysis. It involves collecting, recording, and evaluating the requirements and expectations that help in defining the project's specifications. Both functional and non-functional requirements analysis are included in requirement analysis. The precise behaviours and functionalities that a system must have are referred to as the functional requirements whereas the characteristics that the system must have but do not directly relate to particular functionalities or behaviours are the non-functional requirements.

3.1.1.1 Requirements

Functional requirements are an essential part of the development of software and systems. They define the exact functionalities, features, and capabilities that a system must have in order meet its users' needs. These criteria guide the development process by serving as a road map for designers, developers, and testers.

The functional requirements of the system are as follows:

- To allow customers and employees to register and login to the system.
- To allow customers to request bookings.
- To allow customers to cancel bookings.
- To allow employees to view customer bookings.
- To allow employees to view cancellation predictions.
- To allow employees to update checked in status.

A use case diagram is a type of Unified Modelling Language (UML) diagram that visually shows the interactions between several actors and a system. It gives a high-level overview of the system functionality and how users and external entities interact with it. During the early stages of software development, use case diagrams are frequently used to capture and explain the system's needs.

The following use case diagram for the Hotel Booking Cancellation Prediction System shows the interaction between actors (customer and employee) with the system depicting the functional requirements:

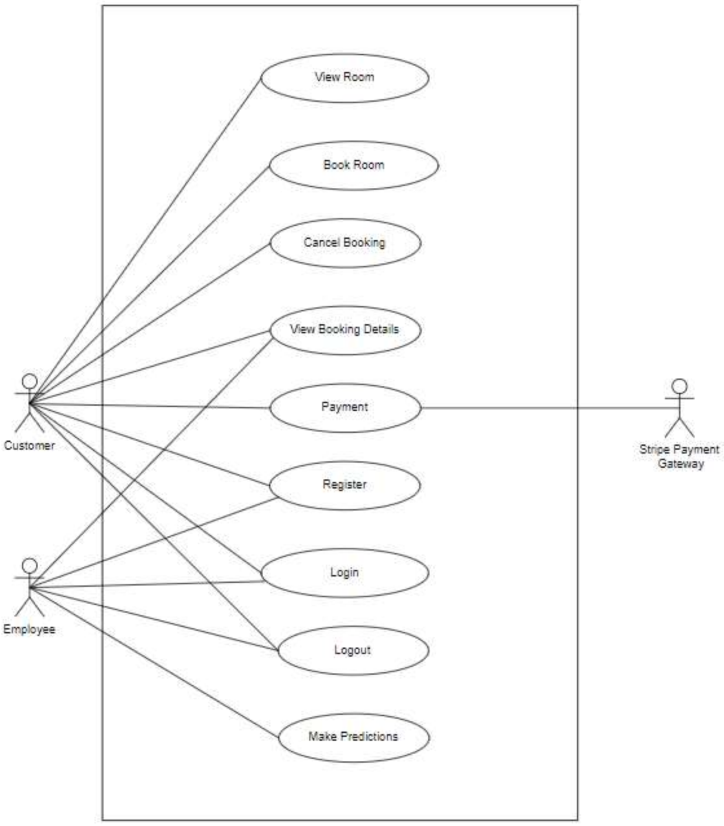


Figure 3.1: Use Case Diagram

Use Case Description:

Here are some of the use case descriptions of the project:

Table 3.1 Use Case Description for Customer Registration

Use Case ID	UC-01
Use Case Name	Customer Registration
Primary Actor	Customer
Secondary Actor	
Description	Registers customer to the system.
Pre-Condition	
Success Scenario	Customer can login to the system. Customer data is stored in the database.
Failure Scenario	Customer is redirected to register again.

Table 3.2 Use Case Description for Customer Login

Use Case ID	UC-02
Use Case Name	Customer Login
Primary Actor	Customer
Secondary Actor	
Description	Logs customer into the system.
Pre-Condition	Customer must be registered to the system.
Success Scenario	Customer is redirected to the home page.
Failure Scenario	Customer is redirected to login again.

Table 3.3 Use Case Description for Booking Request

Use Case ID	UC-03
Use Case Name	Booking Request
Primary Actor	Customer
Secondary Actor	
Description	Customer provides details to request booking.
Pre-Condition	Customer must be logged in to the system.
Success Scenario	Booking request saved to the database and forwarded to the employee.
Failure Scenario	Booking request not saved to the database and not forwarded to the employee.

Table 3.4 Use Case Description for Booking Cancellation Prediction

Use Case ID	UC-04
Use Case Name	Booking Cancellation Prediction
Primary Actor	Employee
Secondary Actor	
Description	System predicts whether a customer is likely to cancel a booking or not.
Pre-Condition	Customer requests must be made and employee must be logged in to the system.
Success Scenario	Prediction result is saved to the database.
Failure Scenario	Prediction result not saved to the database.

3.1.1.2 Non Functional Requirements

Non-functional requirements define how a system should perform its functions rather than what functions it should perform. Non-functional requirements are frequently linked to characteristics like performance, security, usability, and maintainability. They are essential in ensuring that the system adheres to specific standards and limits.

The non-functional requirements of the system are as follows:

- **Security:** The system allows only the registered users to log in to the system and any other users are not allowed to access the system functionality without logging in. Customers and employees in the system are provided with different privilege levels.
- **Usability:** The system has a simple, responsive, and navigable interface that is easy to use.
- **Maintainability:** The system is built using an object-oriented approach which can be easily modified according to changing requirements and the documentation also adds to the maintainability of the system.

3.1.2 Feasibility Analysis

Feasibility study is a thorough investigation that takes place early in the requirement engineering process. The purpose of a feasibility study is to determine the viability of implementing the system. The feasibility study completed for the project is listed below:

3.1.2.1 Technical Feasibility

Technical feasibility refers to the evaluation of a development organisation's or individual's capacity to build a suggested system. Every group member is knowledgeable about the technology and tools used to build the system. The project is therefore technically possible.

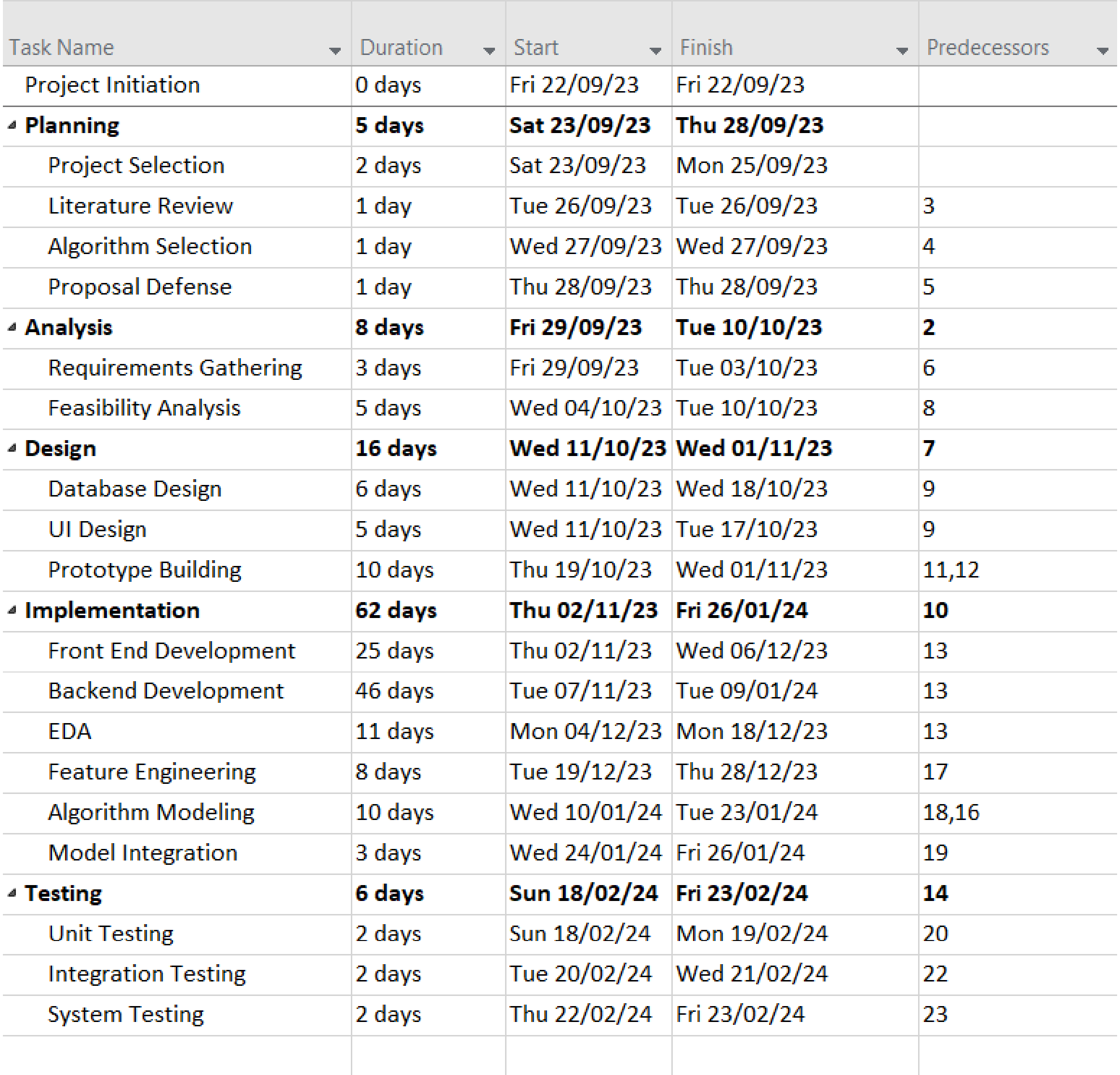
3.1.2.2 Operational Feasibility

Operational feasibility refers to assessing the degree to which a proposed system solves the business problems. The prediction system helps to provide a strategic solution to solve the uncertainties of a booking. The project is therefore operationally possible.

3.1.2.3 Schedule Feasibility

Schedule feasibility is the process of evaluating whether a proposed plan or project can be realistically completed within a specified timeframe. It involves assessing a number of variables, including resource availability, time restrictions, task dependencies, and possible risks.

Given below is the WBS and Gantt chart, which depicts the entire schedule of the project. As the project is successfully completed within the appointed time, it is schedule feasible.



Task Name	Duration	Start	Finish	Predecessors
Project Initiation	0 days	Fri 22/09/23	Fri 22/09/23	
Planning	**5 days**	**Sat 23/09/23**	**Thu 28/09/23**	
Project Selection	2 days	Sat 23/09/23	Mon 25/09/23	
Literature Review	1 day	Tue 26/09/23	Tue 26/09/23	3
Algorithm Selection	1 day	Wed 27/09/23	Wed 27/09/23	4
Proposal Defense	1 day	Thu 28/09/23	Thu 28/09/23	5
Analysis	**8 days**	**Fri 29/09/23**	**Tue 10/10/23**	**2**
Requirements Gathering	3 days	Fri 29/09/23	Tue 03/10/23	6
Feasibility Analysis	5 days	Wed 04/10/23	Tue 10/10/23	8
Design	**16 days**	**Wed 11/10/23**	**Wed 01/11/23**	**7**
Database Design	6 days	Wed 11/10/23	Wed 18/10/23	9
UI Design	5 days	Wed 11/10/23	Tue 17/10/23	9
Prototype Building	10 days	Thu 19/10/23	Wed 01/11/23	11,12
Implementation	**62 days**	**Thu 02/11/23**	**Fri 26/01/24**	**10**
Front End Development	25 days	Thu 02/11/23	Wed 06/12/23	13
Backend Development	46 days	Tue 07/11/23	Tue 09/01/24	13
EDA	11 days	Mon 04/12/23	Mon 18/12/23	13
Feature Engineering	8 days	Tue 19/12/23	Thu 28/12/23	17
Algorithm Modeling	10 days	Wed 10/01/24	Tue 23/01/24	18,16
Model Integration	3 days	Wed 24/01/24	Fri 26/01/24	19
Testing	**6 days**	**Sun 18/02/24**	**Fri 23/02/24**	**14**
Unit Testing	2 days	Sun 18/02/24	Mon 19/02/24	20
Integration Testing	2 days	Tue 20/02/24	Wed 21/02/24	22
System Testing	2 days	Thu 22/02/24	Fri 23/02/24	23

Figure 3.2: Work Breakdown Structure

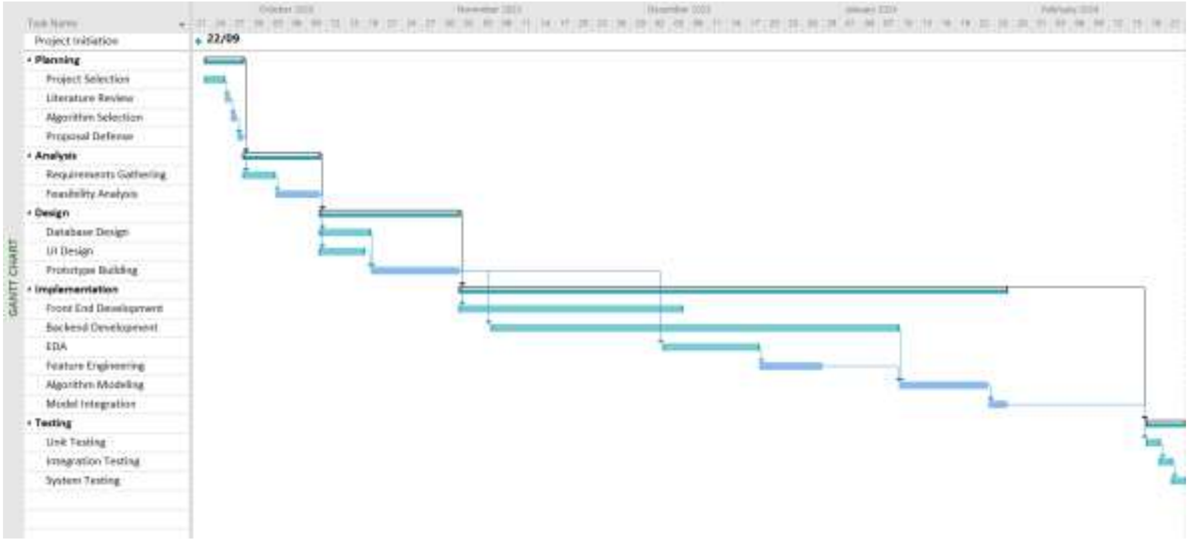


Figure 3.3: Gantt Chart

3.1.3 Analysis

In the analysis phase, requirements of the system are structured. Object oriented approach is used to structure the requirements.

3.1.3.1 Object Modelling

Object modelling is a technique used to represent the structure and interactions of objects within a system. Object modelling can be achieved through the use of object diagrams and class diagrams. Object diagrams provide a snapshot view of the system, showing the objects and their relationship during runtime.

The following object diagram shows the objects of the system, and their relationships.

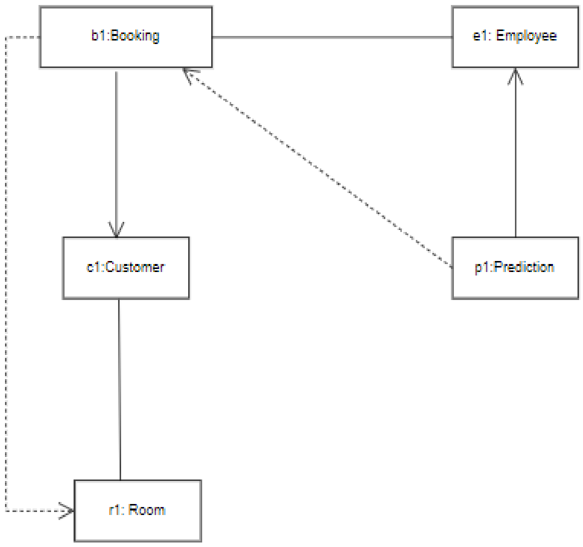


Figure 3.4: Object Diagram

CHAPTER 4: SYSTEM DESIGN

4.1 Design

The design phase is a crucial stage where the overall structure and architecture of the system are planned and defined. The main focus of this phase is to transform the requirements into a blueprint which can then be used to build the actual software. The class diagram, activity diagram, and sequence diagram are utilised in this project to illustrate the system's entire workflow.

4.1.1 Class Diagram

Class diagrams are structural UML diagrams that describe the overall structure of the system. These diagrams show the different classes, their attributes, their methods and the relationships that exist between them. Class diagrams only depict the static view of the system.

The Hotel Booking Cancellation Prediction System consists of 4 Classes: Customer, Employee, Room, and Booking. Booking and Customer classes have a one-to-many relationship, whereas Booking and Employee classes have a many-to-many relationship. Multiple bookings can be made of the same room, so they have a one-to-many relationship as well. The following class diagram shows the structure of the application.

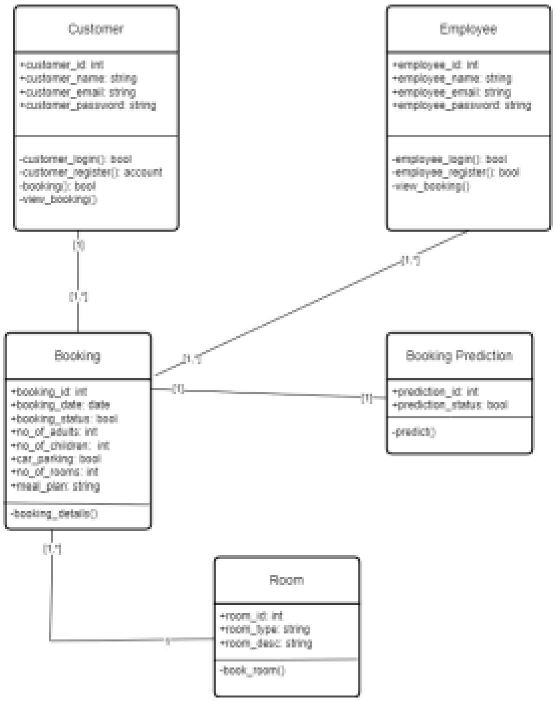


Figure 4.1 Class Diagram

Figure 4.1: Class Diagram

4.1.2 Activity Diagram

Activity Diagrams are behavioural UML diagrams that showcase the flow of activities within a system. Activity Diagrams are used to depict the dynamic aspects of a system and represent the sequence in which the tasks are performed. The following activity diagram shows the process of logging into the application and successfully booking a room.

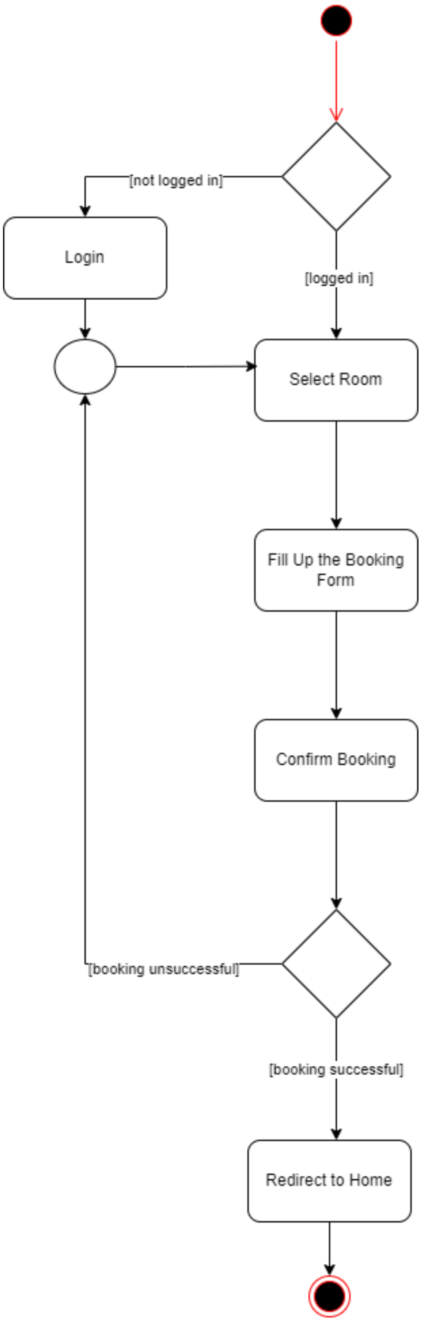


Figure 4.2 Activity Diagram

Sequence Diagram

Sequence diagrams are UML diagrams used to visualise the interaction between different components or objects of the system over time. They describe the sequence of messages exchanged between the components of the system in order to perform a certain task.

The following sequence diagram illustrates the process of making a booking prediction and completing the payment.

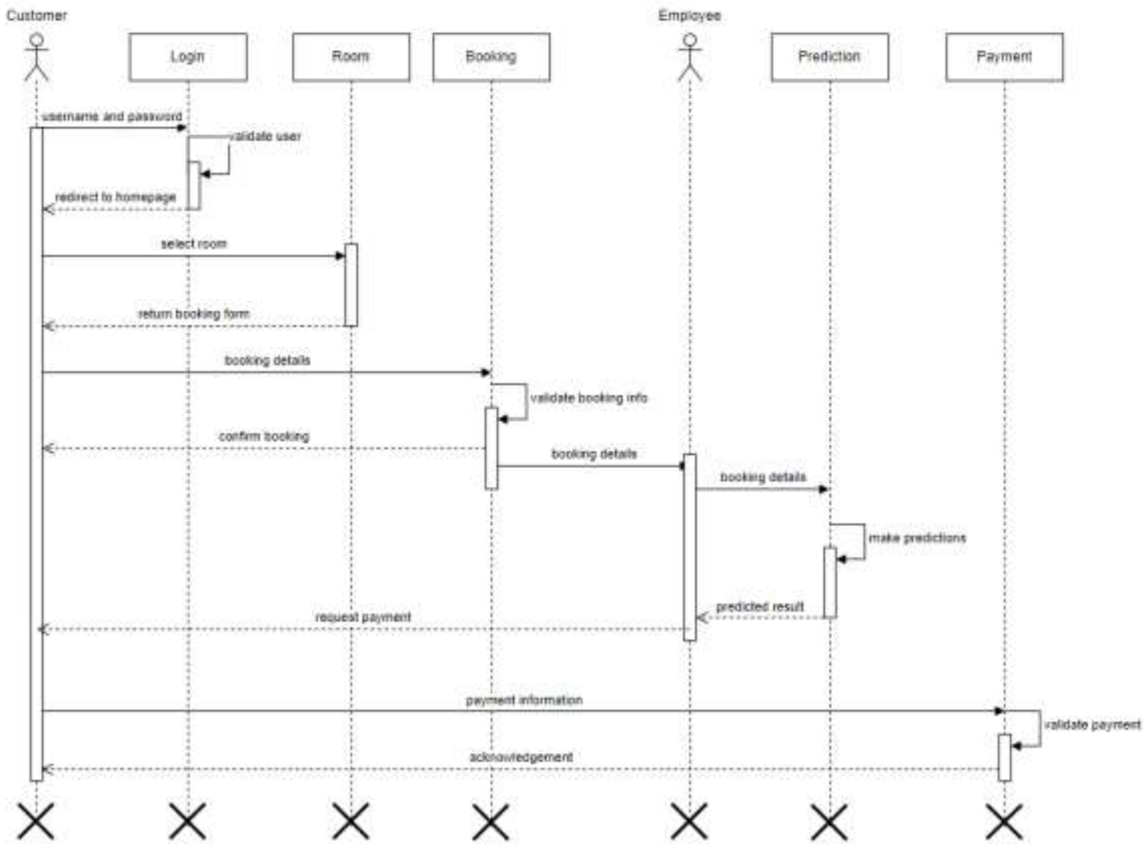


Figure 4.3 Sequence Diagram

4.2 Algorithm Details

The system aims to classify bookings into two categories, likely to cancel and unlikely to cancel. In the context of binary classification, the decision tree efficiently assigns each instance to one of two classes. However, it is sensitive to small variations in training data and may lead to overfitting, so ensemble technique is desirable. The project therefore uses Random Forest Algorithm which in a broader sense is a combination of decision trees.

Following is a basic diagram for decision tree:

4.2.1 Decision Node

Decision nodes partition data into two or more groups based on certain attribute values. Each decision node includes:

- Splitting column name of the dataset through which data items are splitted to left or right node.
- Threshold value for split.
- Left node object reference.
- Right node object reference.
- Impurity metrics of the selected column on the basis of which the data items are splitted to left or right node.
- Result value that is calculated for calculating information gain of each column.

4.2.2 Impurity Metrics

Impurity metrics are used to determine the optimal split column for data at each node of the decision tree. Impurity metrics evaluate the purity of each feature column in a decision node, selecting the best overall value for further splitting. The purpose of impurity metrics is to maximise data uniformity within each partition. In this project, two impurity metrics, entropy and information gain is used which helps to split data points.

4.2.3 Entropy

Entropy is used to find the uncertainty associated with the data. It uses the probability of a given set of values in columns and overall logarithmic measure for calculating impurity over the data columns. It is defined as:

$$H(x) = \sum_{i=1}^{c} \quad -p(i) * log2(p(i))$$

Where,

i is feature name indices,

c is total number of columns in the dataset,

x is the dataset used for training.

Since, entropy is a probabilistic measure its value ranges from 0 to 1. The value of entropy is 0 when all values at a given data column are of the same class. The value of entropy is 1 when the value of the data column is evenly distributed over all classes.

4.2.4 Information Gain

Information gain assesses the quality of the split at a specific decision node based on the quantity of information gained by each column with the target variable. Here, the columns that have more information to the target variable are chosen based on their frequency. Information gain not only considers the quality of the split but also the total size of each data split and the size of outcomes for each data split. It is defined as:

$$Information\ Gain(x, A) = H(x) - \sum_{v \epsilon Values(A)} \frac{|Sv|}{|S|} * H(Sv)$$

Where,

A is a column name,

S is the probability of given target outcomes,

Sv is the probability of given target outcomes with the set of feature column values.

4.2.5 ID3 Algorithm

The ID3 algorithm is a classification algorithm that uses a greedy approach to create a decision tree for data classification. This involves identifying the columns with the lowest entropy or most information gain. In the decision tree, node represents a feature attribute of the dataset, branch represents the decision rule or decision condition for the given feature attribute of the dataset, and leaf represents the outcome or class. In the project there are 15 different attributes, number of children, number of adults, number of weekend nights, number of week nights, type of meal plan, required car parking space, room type, arrival month, arrival date, repeated guest, number of previous cancellations, number of previous bookings not cancelled, number of special requests, lead time, and average price per room.

These attributes are split to form a decision tree where the decision tree has maximum depth of 10 and minimum samples to split of 100.

Following are the steps of ID3 algorithm:

Input: Dataset (S), Maximum Depth (L), Minimum Samples to Split (M)

Step 1: Create a root node with all the datasets (here, 15 features along with 1 label) . Set Current Level (l) to 0.

Step 2: Find the best input feature to split the dataset using impurity metrics.

Step 2.1: Calculate overall entropy of the given dataset with output labels, $H_y(S)$

Step 2.2: For all input features (X)

Step 2.2.1: Calculate entropy of given column (X_i), $H_{x_i}^{value}(S)$

Step 2.2.2: Calculate information gain of (X_i), $Information\ Gain(S, x_i)$

Step 2.3: Compare information gain of all columns.

Step 2.4: Select maximum information gain column as best split.

Step 3: Divide given S with the selected column as the split.

Step 4: Create child nodes based on given split value and threshold value. Left node is a dataset with a smaller value than threshold, and the right node is a dataset with a larger value.

Step 5: Increase l by 1. Check M with the number of datasets, L with l. Such that if not satisfied, stop the iteration.

Step 6: Repeat Step 2 to Step 4, until leaf node is found.

4.2.6 Ensemble Learning

Ensemble learning is a technique in machine learning that involves combining two or more weak models of the same type or different type with low accuracy for predicting overall performance of a system. In the project, we have used the bagging technique of ensemble learning. It is also referred to as bootstrap aggregation as bagging involves two different steps, bootstrapping and aggregation. Bootstrapping is the technique of sampling which splits the overall data into multiple samples with randomised selection of data with replacement. Aggregation on the other hand, is the process of combining the results provided by independently trained models generated from bootstrapped data. Aggregation combines those data by either taking maximum value or minimum value or taking an average of the overall result. In the project, the final model predicts the output class by taking majority voting of the output class labels, which is also known as hard voting.

4.2.7 Random Forest Algorithm

Random Forest is an ensemble learning algorithm which uses two or more decision trees to form voting classifiers for providing accurate and robust results to the real-world data. In the project, the random forest consists of 10 decision trees.

Following are the steps of random forest algorithm:

Input: Dataset (S), Maximum Depth (L), Minimum Samples to Split (M), Number of trees (N)

Step 1: Randomly samples S into $S_1, S_2, \dots, S_N$ and create N different trees into $T_1, T_2, \dots, T_N$

Step 2: For each decision tree T_i, use ID3 algorithm for given bootstrapped samples S_i with M as maximum samples to split and L as maximum depth of the decision tree.

Step 3: For given set of input data X, predict the output (P_i) with the help of trees.

Step 4: Use hard voting method for aggregating the data into the output prediction P.

$$P = max\ (P_1, P_2, \dots, P_N)$$

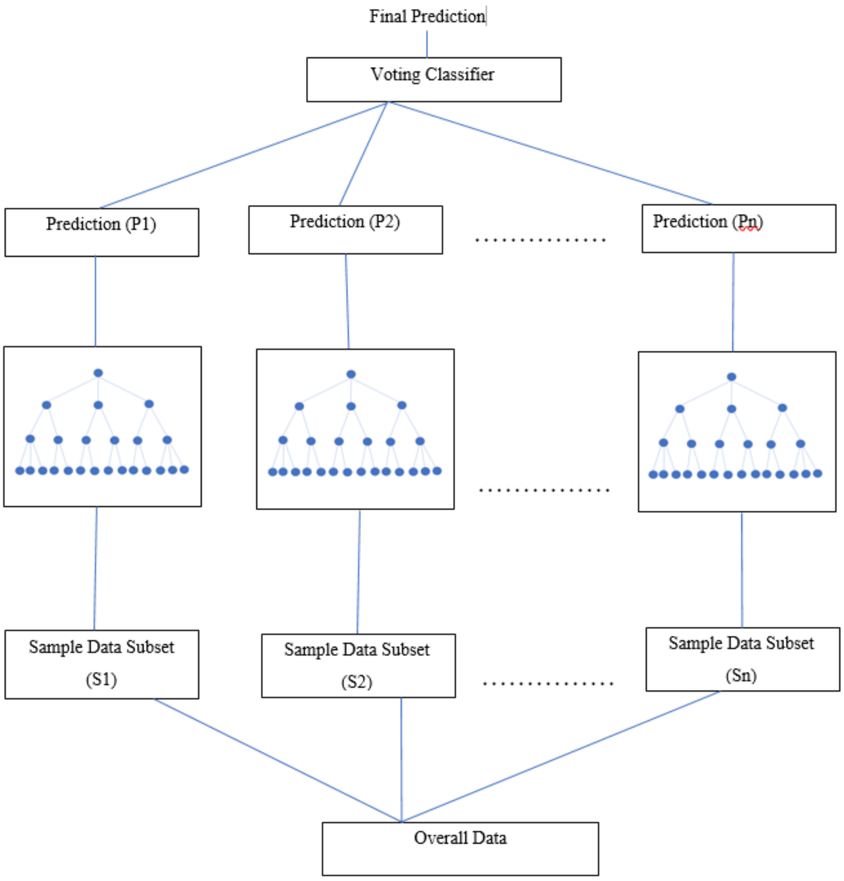


Figure 4.4 Random Forest Classifier

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation

Implementation is the process that includes building an operational system through the efficient application of suitable tools and technologies that are appropriate for the given project. This process converts established designs into a functional system.

5.1.1 Tools used

The following are the tools used in the implementation of project and the making of document:

Front-end Tools

Front-end tools are those tools that enable the implementation of an application so that users can communicate with the system directly. A system's front-end acts as an intermediary between the user and the back-end. The following front-end tools are used in the development of this project:

HTML and CSS

The mark-up language used to organise the content of a web page is called HTML. Creating web pages mostly requires the use of HTML and CSS. The style sheet created for displaying web pages is called CSS. The web pages are presented using HTML elements. The ability to separate presentation from HTML allows web pages to have the right layout, font for text, and colour scheme.

JavaScript

The language used for client-side scripting is JavaScript. JavaScript is used for making content dynamic and fetching data from the server asynchronously.

Bootstrap

Bootstrap is a front-end framework which incorporates HTML, CSS, and JavaScript for developing responsive websites. In this project, bootstrap is used for developing the website.

Back-end tools

Back-end tools are those tools that enable the system to function properly when a user visits a web page. An application's back-end gives the system's front-end the resources and data access it needs to respond to requests. The project is developed using the following back-end tools:

Python (v3.11.5)

Python is a general-purpose programming language used for scripting. Python scripts are used in the back end to integrate machine learning models, populate databases for visualisation, and interact with web pages through templating.

Django (v4.2.5)

Django is a free and open-source web development framework that enables complex and secure applications. It uses server-side MVT architecture, comparable to MVC design. Django is used to provide dynamic behaviour to an application.

SQLite

SQLite is a disk-based RDBMS that is lightweight and suitable for online applications. It is a file-based database system that may be easily accessed using SQL commands for data definition, manipulation, and other related tasks. Because it is file-based, it can be readily embedded into a program and does not require any form of engine to access data.

Visual Studio Code

Visual Studio Code is a lightweight, cross-platform, and open-source IDE for creating various applications. It offers a wide range of functionality for the development of projects.

Algorithm Development Tools

The following are the tools used for the development and deployment of algorithm to the system:

Jupyter Notebook

Jupyter Notebook is an interactive, open-source web platform for analysing and developing various applications. It facilitates data visualisation and collaboration with team members, allowing for faster project development.

Git and GitHub

Git is a software configuration management solution that allows for distributed software version control. Git was used to track feature changes and ensure effective team collaboration. GitHub is a web-based service, providing hosting for git repositories. GitHub offers services for managing pull requests, tracking issues, and conducting code reviews.

CASE Tools

CASE tools are used to analyse and develop a project. They are used to design the essential diagrams and provide various project artefacts. The following are the CASE tools used in the project:

Draw.io

Draw.io is a free and open-source CASE tool available across multiple platforms. It is used to create a variety of wireframes and diagrams, including object, class, use case, activity, and sequence diagrams.

Microsoft Project

Microsoft Project is a project management tool. This project utilises Microsoft Project for task breakdown, resource allocation, scheduling, and risk management.

Dependencies

Dependencies are the libraries required for project implementation and operation. The dependencies used in the project are the external libraries and packages developed by third party developers or community. These external modules must be downloaded separately.

The following are the dependencies or external modules used in the project:

Table 5.1 External Modules

S.No.	Modules	Description
i.	numpy	This package helped to perform mathematical operations on data.
ii.	pandas	This package helped to manipulate dataframes.
iii.	sklearn	This package helped to preprocess the data.

iv.	matplotlib & seaborn	These packages helped to visualise the data.

5.1.2 Implementation Details

The overall project is divided into three different components, front-end, back-end, and algorithm implementation for development. The following figure shows the implementation process of the algorithm in detail.

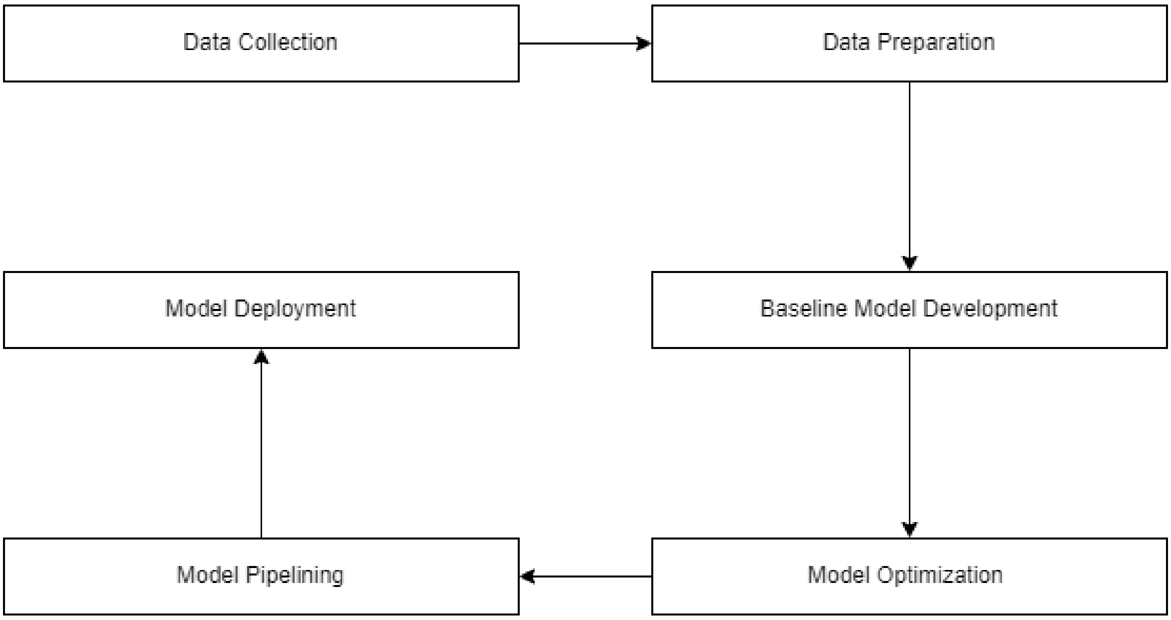


Figure 5.1 Implementation Process

5.1.2.1 Data Collection

For the data used in the project, it was downloaded from publicly available repository, Kaggle. The data originally consisted of 36275 unique records, and 18 different features out of which 15 features were chosen for the project.

Table 5.2 Data Columns

Data Column	Data Type	Range/ Value
no_of_adults	Integer	1-10
no_of_children	Integer	1-10
no_of_weekend_nights	Integer	0-7
no_of_week_nights	Integer	0-17

type_of_meal_plan	Categorical	Meal Plan 1, 2, 3, and Not Selected
required_car_parking_space	Binary	0 ,1
room_type_reserved	Categorical	Room Type 1,2,3,4,5,6, and 7
arrival_month	Integer	1-12
arrival_date	Integer	1-31
repeated_guest	Binary	0,1
no_of_previous_cancellation	Integer	0-13
no_of_previous_bokings_not_cancelled	Integer	0-58
no_of_special_requests	Integer	0-5
lead_time	Integer	0-365
avg_price_per_room	Float	50-540

5.1.2.2 Data Preparation

Data preparation is a critical phase that involves cleaning, transforming, and organising raw data so that it is suitable for analysis and model building. Proper data preparation guarantees that the information utilised in analysis and model building is correct,complete, and relevant.

The following are the different procedures used during data preparation:

1. **Exploratory Data Analysis**

   Exploratory data analysis helps in gaining the overall information of the data set being used for a particular problem statement. In performing EDA for the project, various insights were gained. It was found that the dataset had no missing values as a result data imputation was not required.

   Firstly, feature selection was performed and only the relevant features considering the project scope were considered dropping all the other irrelevant columns. The Booking_ID column was removed because of high cardinality and also because it had no relation with the target variable. Similarly, market_segment_type and arrival_year columns were not relevant to the scope of the project and were dropped. Noises like no_of_adults having value 0, avg_price_per_room having value 0 were removed. Box-plot helped in finding outliers present in lead_time, and were removed.

Different visualisation techniques helped to gain knowledge about the data and their relationship with one another. Count plot on target variable helped to find out that the percentage of cancellations is 33.46% and that of non-cancellations is 66.54% in the dataset. This indicates that there is some imbalance in the target variable, however it is not highly imbalanced. Distribution plot helped to discover that lead_time and avg_price_per_room are right skewed.

Correlation analysis of various features with the target variable helped to discover that lead_time, no_of_special_requests, avg_price_per_room have comparatively high correlation and other features have low correlation.

2. **Train Test Split**

After processing the original dataset, it was reduced to contain 35019 records. These 35019 instances were split into two sets of data, training set and test set with 20% test size. The training set is used to train the model, and the test set is used to evaluate its performance.

Table 5.3 Dataset Split Size

Dataset	Size
Training Features	(28015, 15)
Training Labels	(28015,1)
Test Features	(7004,15)
Test Labels	(7004,1)

3. **Scaling**

The dataset has various ranges for each column. Scaling data maintains its standard format for all the data columns. This is done for the dataset's numerical data columns. This project follows the standard scaling process.

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Here,

x_{scaled} is the scaled value,

x is the original value,

μ is the mean of the data feature column,

σ is the standard deviation of the data feature column.

The columns lead_time and avg_price_per_room were passed for standard scaling.

4. **Encoding**

   Encoding is a preprocessing technique that converts categorical values into numerical values. There are one-hot encoding and label encoding techniques for encoding. In the project, label encoding is used to convert type_of_meal_plan (0-3) , room_type_reserved (0-6) , and booking_status (0,1) columns.

5.1.2.3 Baseline Model Development

A baseline model provides a simple and initial answer to the problem at hand, and it serves as a reference point for evaluating the performance of complex models. Decision tree with minimum samples to split of value 2, and maximum depth until all leaves prune served as the baseline model of the project.

The following is the performance result of the baseline model:

Table 5.4 Baseline Model Metrics

Dataset	Imbalanced Accuracy	Balanced Accuracy	AUC Score	Macro Average Precision	Macro Average Recall	Macro Average F1-Score
Training	78.73	64.72	64.72	0.77	0.65	0.67
Testing	75.65	60.91	60.91	0.71	0.61	0.62

5.1.2.4 Model Optimisation

Optimising a model involves adjusting hyperparameters to improve its performance. Optimising a machine learning model involves adjusting multiple parameters to achieve optimal performance, leading to increased complexity. In the project, random forest is used to increase the complexity and improve model performance. The purpose of model optimization is to identify the optimum hyperparameters and improve the model's performance. Hyperparameter tuning is used for model optimization in the project. Hyperparameter tuning optimises an algorithm's performance by iteratively selecting hyperparameters to ensure a robust algorithm. There are different techniques to achieve hyperparameter tuning, and in the project grid search approach was used to identify the hyperparameters.

Grid search is a hyperparameter tuning process that requires defining all potential combinations of hyperparameters. Grid search is often computationally expensive, but it produces an ideal model. The hyperparameters for decision trees are minimum samples to split, maximum depth, and that for random forests are minimum samples to split, maximum depth, and number of trees. The best estimators were evaluated with the help of grid search, and were found to be 10 for number of trees, 11 for maximum depth and 100 for minimum samples to split.

5.1.2.5 Model Pipelining

Model Pipelining is the process of developing a step-by-step technique for interconnected data preparation. It combines data preprocessing, feature extraction, and model training in a single procedure.The pipeline moves through a sequence of models, each of which addresses a distinct subtask. The model pipelining approach simplifies the model training process, allowing it to be easily automated. Overall, model pipelining provides a disciplined framework for tackling complex machine learning problems.

The following is the model pipeline of the system:

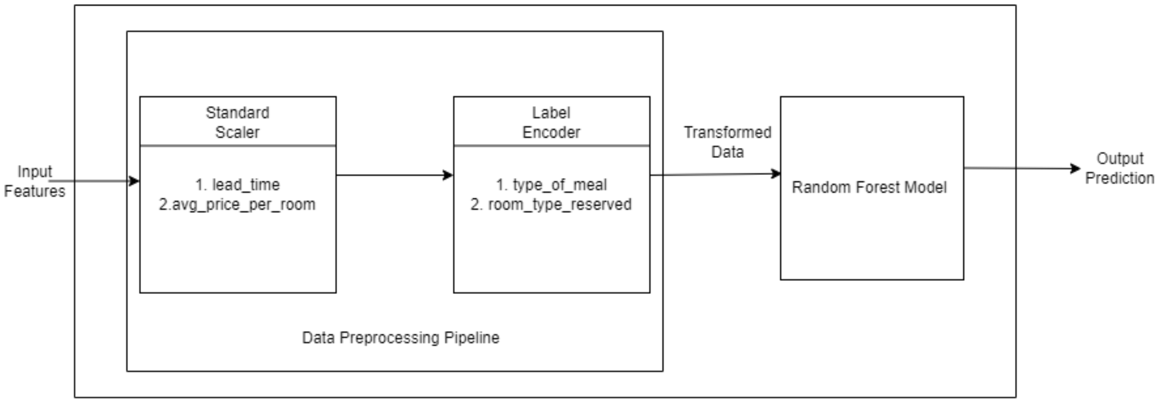


Figure 5.2 Model Pipelining

5.1.2.6 Model Deployment

After pipelining, the model was transformed to a pickled format using pickle. Pickling is the process of converting an object into a byte stream, while unpickling is the inverse procedure of converting a byte stream back into the original object. The pickled model was then implemented into the web app. The hyperparameters chosen when optimising the model in model pipelining were frozen for prediction purposes. The customer's booking information is correctly sent to the frozen model, enabling it to forecast the result with accuracy.

5.2 Testing

Testing is the process of verifying the functionality, quality, and performance of the system in hand. Testing is carried out throughout the development of a system. Testing involves thoroughly evaluating the system with the intent of identifying any bugs or errors that might be present and ensuring that the system meets the required specifications and standards.

Unit testing, followed by system testing were carried out to test the functionality of this system.

5.2.1 Unit Testing

Unit testing is a software testing technique where each individual component is tested independently to the rest of the system. The main goal of unit testing is to verify that each unit performs as expected, without any errors.

Test Cases

Table 5.5 Test Case for Customer Registration

Test Name	Customer Registration	Test Case ID	T01
Test Case Description	Verify Customer Registration Functionality	Test Priority	High
Prerequisite	Provide Email, Password, Confirmation Password, Full Name, and Phone Number	Post Condition	Customer is registered into the database and automatically logged in to the system.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the customer registration page.		Customer should be redirected to the registration page.	Customer is redirected to the registration page..	PASS
2	Verification of customer’s data	1.Enter the customer's email, full name, password, confirmation password, and phone number. 2.Click the ‘Register’ button.	Customer should be successfully registered, and automatically logged in to the application	Customer is successfully registered, and logged in.	PASS
3	Verification of customer’s data when passwords don’t match.	1.Enter the customer’s email, full name, password, confirmation password, and phone number. 2.Click the	Error prompt: “two password fields do not match”	Error prompt: “two password fields do not match”	PASS

		‘Register’ button.			
4	Verification of customer’s data when an existing email is entered	1.Enter the customer’s email, full name, password, confirmation password, and phone number. 2.Click the ‘Register’ button.	Error prompt: “user with this email already exists”	Error prompt: “user with this email already exists”	PASS

Table 5.6 Test Case for Customer Login

Test Name	Customer Login	Test Case ID	T02
Test Case Description	Verify Customer Login Functionality	Test Priority	High
Prerequisite	Provide Email and Password	Post Condition	Customer is logged in to the system.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the customer login page.		Customer should be redirected to the login page.	Customer is redirected to the login page.	PASS
2	Login using username and password that don’t exist.	1.Enter username = ‘alex’ password = ‘orchid321’ 2.Click ‘Login’ button.	Error prompt: “username or password incorrect”	Error prompt: “username or password incorrect”	PASS
3	Login using correct	1.Enter username =	Error prompt: “username or	Error prompt: “username or	PASS

	username but incorrect password.	‘subrat’ password = ‘nepal@2024’ 2. Click ‘Login’ Button	password incorrect”	password incorrect”	
4	Login using correct username and password	1.Enter username = ‘subrat’ password = ‘boudha@123’ 2. Click ‘Login’ Button	Customer should be logged in and redirected to the home page.	Customer is logged in and redirected to the home page.	PASS

Table 5.7 Test Case for Booking a Room

Test Name	Booking a Room	Test Case ID	T03
Test Case Description	Customer books a room.	Test Priority	High
Prerequisite	Select a room, and fill up the booking form.	Post Condition	Customer successfully books a room.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to Booking Form..	1.Navigate to the ‘Rooms’ Page. 2.Select a room.	Customer should be shown the booking form.	Customer is shown the booking form.	PASS

2	Book a room with invalid values.	1.Enter ‘Number of Adults’ = 0, ‘Number of Children = -1’, ‘Check-in Date: 20-10-2011’ ‘Check-out Date: 19-20-2011’ 2.Click ‘Book Room’ button.	Error prompt: “Number of Adults can’t be less than 1.” “Number of Children can’t be less than 0.” “Check-in date cannot be in the past” “Check-out date cannot be before the check-in date”	Error prompt: “Number of Adults can’t be less than 1.” “Number of Children can’t be less than 0.” “Check-in date cannot be in the past” “Check-out date cannot be before the check-in date”	PASS
3	Book a room with valid data.	1.Enter ‘Number of Adults’ = 2 ‘Number of Children = 0’, ‘Check-in Date: 20-05-2024’ ‘Check-out Date: 30-05-2011’ 2.Click ‘Book Room’ button.	Room should be successfully booked.	Room is successfully booked	PASS

Table 5.8 Test Case for Employee Registration

Test Name	Employee Registration	Test Case ID	T04
Test Case Description	Verify Employee Registration Functionality	Test Priority	High
Prerequisite	Provide Email, Password, Confirmation Password, Full Name, and Phone	Post Condition	Employee is registered into the database and automatically logged in to the

	Number		system.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the employee registration page.		Employee should be redirected to the registration page.	Employee is redirected to the registration page..	PASS
2	Verification of employee’s data	1.Enter the employee's email, full name, password, confirmation password, and phone number. 2.Click the ‘Register’ button.	Employee should be successfully registered, and automatically logged in to the application	Employee is successfully registered, and logged in.	PASS
3	Verification of employee’s data when passwords don’t match.	1.Enter the employee’s email, full name, password, confirmation password, and phone number. 2.Click the ‘Register’ button.	Error prompt: “two password fields do not match”	Error prompt: “two password fields do not match”	PASS

4	Verification of employee's data when an existing email is entered	1.Enter the employee's email, full name, password, confirmation password, and phone number. 2.Click the 'Register' button.	Error prompt: "user with this email already exists"	Error prompt: "user with this email already exists"	PASS

Table 5.9 Test Case for Employee Login

Test Name	Employee Login	Test Case ID	T02
Test Case Description	Verify Employee Login Functionality	Test Priority	High
Prerequisite	Provide Email and Password	Post Condition	Employee is logged in to the system.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the employee login page.		Employee should be redirected to the login page.	Employee is redirected to the login page.	PASS
2	Login using username and password that don't exist.	1.Enter username = 'adam' password = 'iamadam123' 2.Click 'Login' button.	Error prompt: "username or password incorrect"	Error prompt: "username or password incorrect"	PASS
3	Login using correct username but incorrect password.	1.Enter username = 'supriya' password = 'hotelier120'	Error prompt: "username or password incorrect"	Error prompt: "username or password incorrect"	PASS

		2. Click ‘Login’ Button			
4	Login using correct username and password	1.Enter username = ‘supriya’ password = ‘hotelier123’ 2. Click ‘Login’ Button	Employee should be logged in and redirected to the home page.	Employee is logged in and redirected to the home page.	PASS

Table 5.10 Test Case of Booking Prediction

Test Name	Booking Prediction	Test Case ID	T06
Test Case Description	Predict booking’s cancellation status.	Test Priority	High
Prerequisite	Booking must have been made.	Post Condition	Booking is predicted as ‘Likely to Cancel’ or ‘Unlikely to Cancel’

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the pending bookings page.		Employee should be shown a list of pending bookings.	Employee is shown a list of pending bookings.	PASS
2	Predict the Booking Status.	1.Select a Booking. 2.Click ‘Predict’ Button.	Classified as ‘Likely to Cancel’ or ‘Unlikely to Cancel’	Classified as ‘Likely to Cancel’	PASS

Table 5.11 Test Case for Booking Payment

Test Name	Booking Payment	Test Case ID	T07
Test Case Description	Customer performs payment through ‘Stripe’ payment gateway.	Test Priority	High
Prerequisite	Booking status must already be predicted.	Post Condition	Payment must be completed for the booking.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Navigate to the pending payments page.		Customer should be shown a list of bookings with pending payments.	Customer is shown a list of bookings with pending payments.s.	PASS
2	Perform payment with invalid data	1.Select ‘Proceed to Payment’ Button 2. Enter invalid credit card information ‘1293 8314 8237’ 3.Click ’Pay’ Button	Error prompt: ‘Your card number is invalid’	Error prompt: ‘Your card number is invalid’	PASS
3	Perform payment with valid data	1.Select ‘Proceed to Payment’ Button 2. Enter invalid credit card information ‘1234 1234 1234 1234’ 3.Click ’Pay’ Button	Successful Payment	Successful Payment	PASS

5.2.2 System Testing

System Testing is a software testing process that evaluates the functionality and behaviour of the entire system. System testing focuses on testing the end-to-end functionality of the system against the specified requirements. This process verifies whether the system meets the functional as well as the non-functional requirements, and performs well in the business environment.

Table 5.12 Test Case for System Testing

Test Name	System Testing	Test Case ID	T08
Test Case Description	Overall System Testing	Test Priority	High
Prerequisite	Registered Customer and Employee.	Post Condition	Successful Booking Prediction and Payment.

Test Steps:

SN	Action	Test Steps	Expected Output	Actual Output	Test Result
1	Booking Prediction	1. Booking is done by a registered user. 2.Employee views the booking. 3.Click on ‘Predict Button’	Booking predicted as “Likely to Cancel” or “Unlikely to Cancel’	Booking predicted as “Unlikely to Cancel”	PASS
2	Perform Payment	1.Navigate to pending payments page. 2.Click ‘Proceed to Payment’ 3.Enter credit card information. 4.Click ‘Pay’	Payment successful	Payment successful	PASS
2	Operational Testing	Enter website URL.	Website launches with expected components.	Website runs as expected.	PASS

			Users can interact with the website properly.		
3	Functionality Testing	Perform operations on the website.	Operations like Login, Registration, Booking, and Prediction work as intended.	All the operations are performed without errors.	PASS
4	Usability Testing	Users are informed of how the system works.	The system must be easy to operate.	The system is easy to operate by users of all technical levels.	PASS

5.3 Result Analysis

Result Analysis is the stage where the performance of a trained model is evaluated and analysed to assess its effectiveness in making accurate predictions. Result analysis is crucial in understanding how well a model is able to generalise unseen data and whether it meets the expected results.

The following evaluation metrics were adopted in measuring the performance of the model:

5.3.1 Accuracy

Accuracy is a metric that measures the proportion of correctly classified instances out of the total instances. Accuracy provides a general indication of the model's correctness across all classes. Accuracy is a useful metric when classes in the dataset are balanced, however, accuracy can be misleading when there is class imbalance.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Number\ of\ Correct\ Predictions}}{\boldsymbol{Total\ Number\ of\ Predictions}}$$

5.3.2 Balanced Accuracy

Balanced Accuracy is a metric used to measure the performance of a classifier, particularly used when two classes are imbalanced.Balanced Accuracy is calculated as the arithmetic mean of Sensitivity (true positive rate) and Specificity (true negative rate).

$$Balanced\ Accuracy\ = \frac{Sensitivity + Specificity}{2}$$

$$Sensitivity\ = \frac{True\ Positive}{True\ Positive\ + False\ Negative} * 100$$

$$Specificity\ = \frac{True\ Negative}{False\ Positive\ + True\ Negative} * 100$$

5.3.4 AUC-ROC Curve

AUC - ROC curve is a performance metric for classification models, especially in scenarios where there is class imbalance. ROC is a probability curve that describes the trade-off between true positive rate and false positive rate, whereas the AUC score measures the performance of the classifier by calculating area under the ROC curve. Higher the AUC Score, the better the model is at correctly predicting the classes.

5.3.5 Macro Average Precision Score

Macro Average Precision Score is a performance metric that calculates the arithmetic mean of individual classes' precision. Precision measures the proportion of true positive predictions out of all positive predictions made by the model.

$$Macro\ Average\ Precision\ = \frac{1}{N}\sum_{i\ =\ 1}^{N}\quad Precision_{\mathrm{i}}$$

Where,

N is the number of classes,

$$Precision\ = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives}$$

5.3.5 Macro Average Recall Score

Macro Average Recall Score is a performance metric that calculates the arithmetic mean of individual classes' recall. Recall measures the proportion of true positive predictions out of all actual positive instances.

$$Macro\ Average\ Recall\ = \frac{1}{N}\sum_{i=1}^{N}\quad Recall_{\mathrm{i}}$$

Where,

N is the number of classes,

$$Recall\ = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives}$$

5.3.6 Macro Average F-1 Score

Macro Average F-1 Score is a performance metric that calculates the F-1 score for each class individually and then computes the unweighted average of those scores. F1 Score is the harmonic mean of precision and recall. It is particularly useful when each class has equal importance, and there is no class imbalance.

$$Macro\ Average\ F1\ Score\ = \frac{1}{N}\sum_{i=1}^{N}\quad F1_{\mathrm{i}}$$

Where,

N is the number of classes,

$$F1\ =\ 2 * \frac{Precision * Recall}{Precision + Recall}$$

The following is the result obtained by hyperparameter tuned random forest model:

Table 5.13 Performance Metrics for Final Model

Dataset	Imbalanced Accuracy	Balanced Accuracy	AUC Score	Macro Average Precision	Macro Average Recall	Macro Average F1-Score
Training	86.56	82.26	0.822	0.87	0.82	0.84
Testing	86.30	81.84	0.818	0.86	0.82	0.84

The hyperparameter tuned model selected to integrate with the web application provided a balanced accuracy of 82.26% during training and 81.84% during test without much variation between test and train which is acceptable.

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

6.1 Conclusion

Hotel Booking Prediction System is a system that provides hoteliers a systematic approach to predicting cancellations of bookings made by customers. This system employs the Random Forest Classifier in predicting whether a booking is likely to be cancelled or not. Customers provide their booking details, which can be viewed by the employees. Predictions are then made by the employee, which can result in one of two classes, “Likely to Cancel” or “Unlikely to Cancel”. The model correctly classified 4461 non cancellations out of 4702, and 1584 cancellations out of 2302 from the testing dataset. Hotels can formulate new policies based on these predictions to minimise cost and maximise profits. Improved revenue management, optimal resource allocation, and reduced overbooking are some of the areas where this system can be of great assistance.

6.2 Future Recommendation

A few things can be included in the future work for even better results. Although this system specifically uses bootstrapping, other ensemble learning techniques like boosting and stacking are also viable options. Additionally, to improve the overall system's evaluation metrics, sophisticated methods such as Neural Networks, Gradient Descent, Batch Normalisation, Regularisation, can be used. Along with using complex models, different real-world features can be additionally considered for more accurate and relevant results. For the enhancement of the overall system, different functionalities can be added such as customer reviews, room availability tracking, recommendations based on customer interests for robust hotel management and prediction systems.

REFERENCES

[1] A. J. S.-M. M. P. Eleazar C. Sanchez, “Identifying Critical Hotel Cancellations using Artificial Intelligence,” International Journal of Hospitality Management, 2020.

[2] Y. Lin, “Research on the Influencing Factors of Cancellation of Hotel,” Department of Automation and electrical engineering, Jinan University, Shandong, 2023.

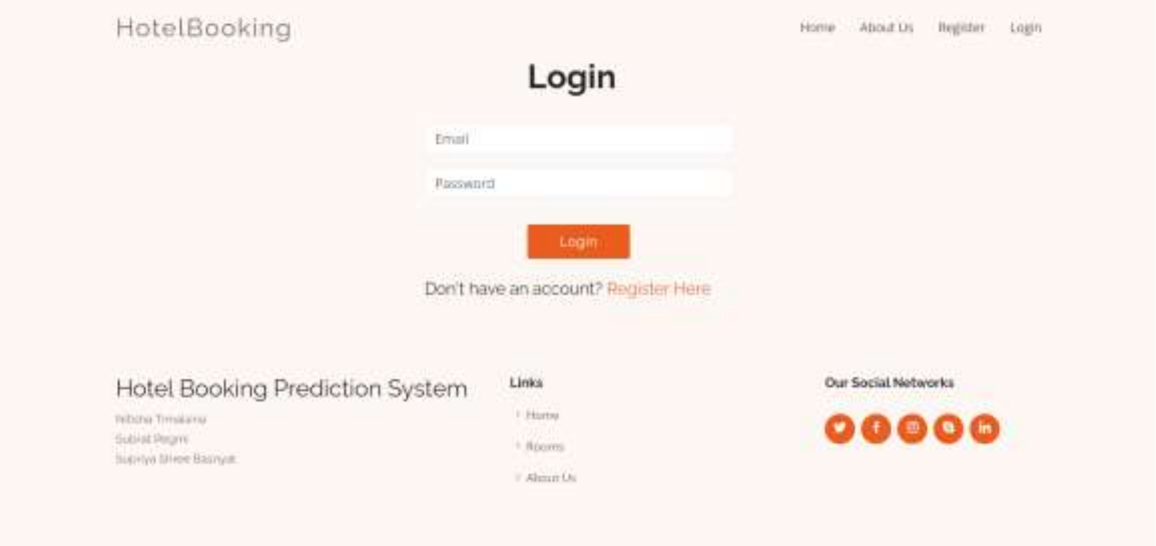
[3] R. Mytnik, “Predicting Hotel Booking Cancellations,” *Analytics Vidhya,* 2021.

[4] S. T. M.R. Martinez-Torresa, “A machine learning approach for the identification of the deceptive reviews,” University of Seville (Spain), 2019.

[5] A. d. A. L. N. Nuno Antonio, “Big Data in Hotel Revenue Management: Exploring Cancellation Drivers to Gain Insights Into Booking Cancellation,” Cornell Hospitality Quarterly, 2019.

APPENDIX

UI Screenshots:



Email

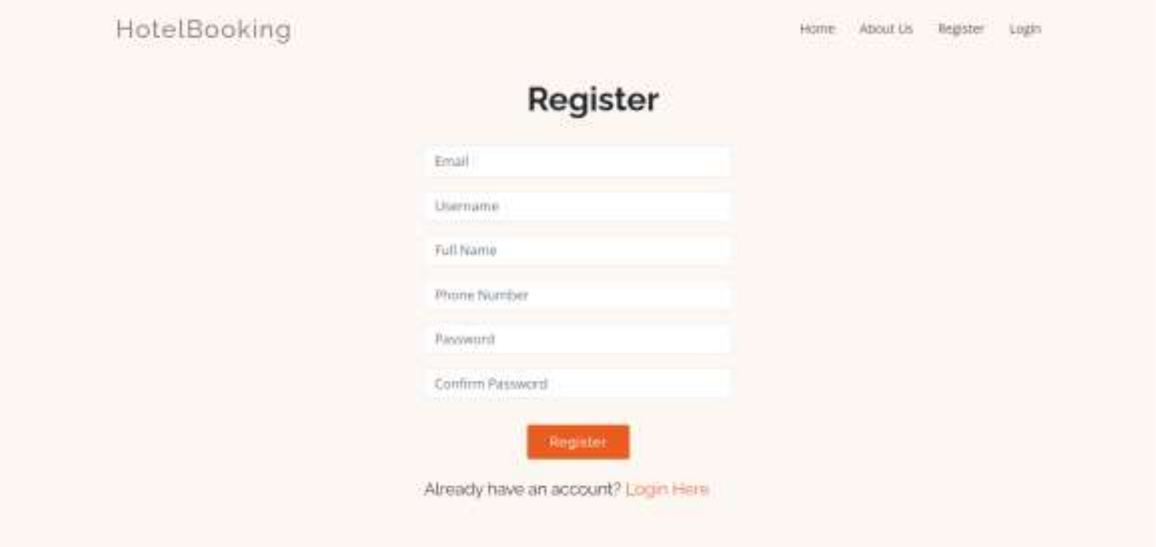
Password

Don't have an account? Register Here

Hotel Booking Prediction System

Nibsha Timalsina
Subrat Regmi
Supriya Shree Basnyat

Appendix A: Login Page



Email

Username

Full Name

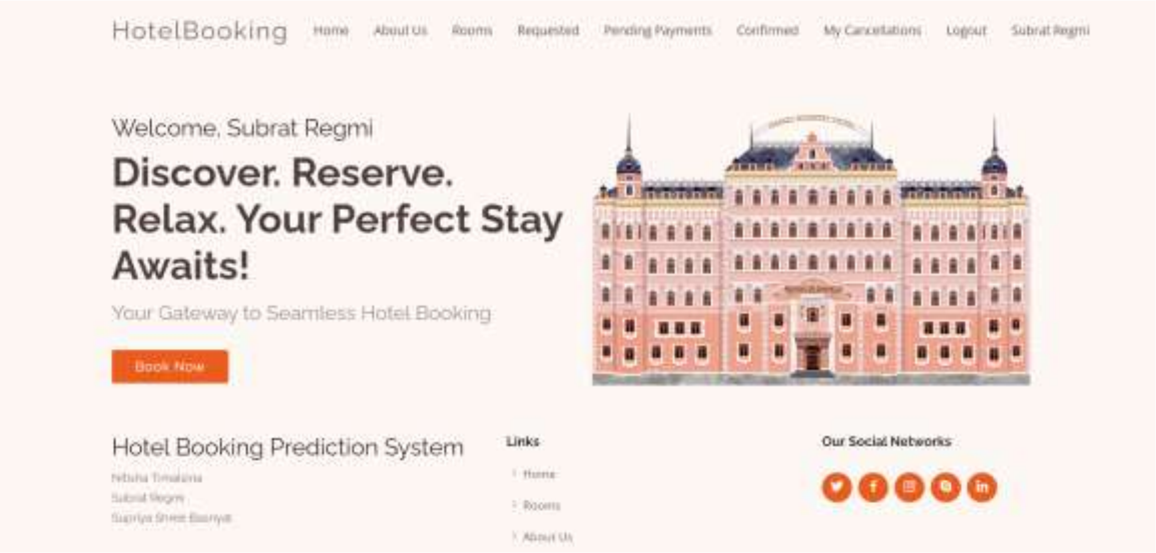
Phone Number

Password

Confirm Password

Already have an account? Login Here

Appendix B: Registration Page



Welcome, Subrat Regmi

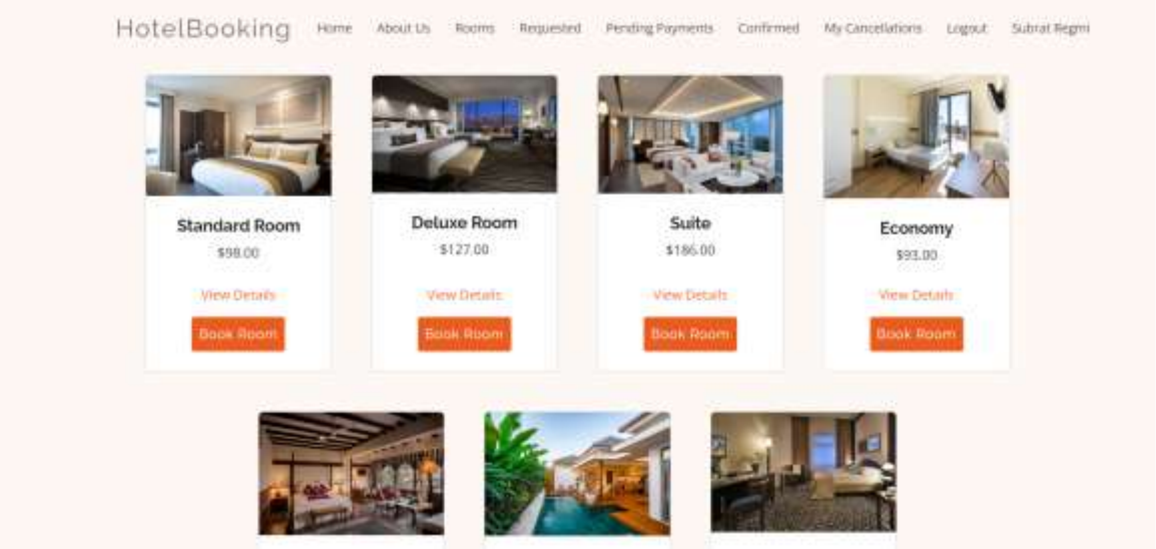
Discover. Reserve. Relax. Your Perfect Stay Awaits!

Your Gateway to Seamless Hotel Booking

Hotel Booking Prediction System

Nitisha Timalsina
Subrat Regmi
Supriya Shree Basnyat

Appendix C: Home Page



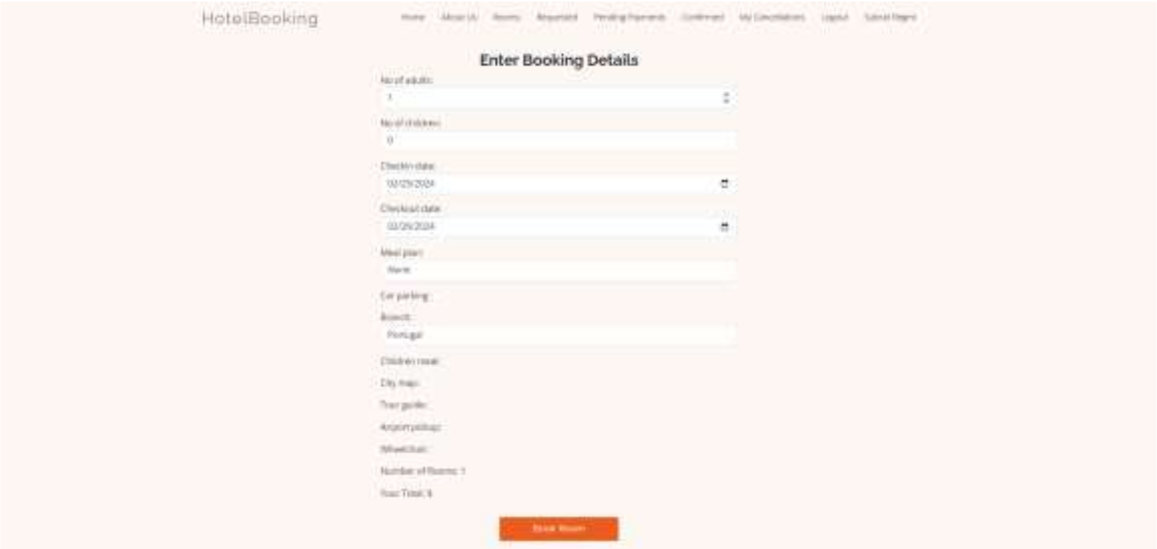
Standard Room
$98.00
View Details
Book Room

Deluxe Room
$127.00
View Details
Book Room

Suite
$186.00
View Details
Book Room

Economy
$93.00
View Details
Book Room

Appendix D: Rooms



Enter Booking Details

No of adults:
1

No of children:
0

Checkin date:
02/25/2024

Checkout date:
02/26/2024

Meal plan:
None

Car parking:

Branch:
Portugal

Children meal:

City map:

Tour guide:

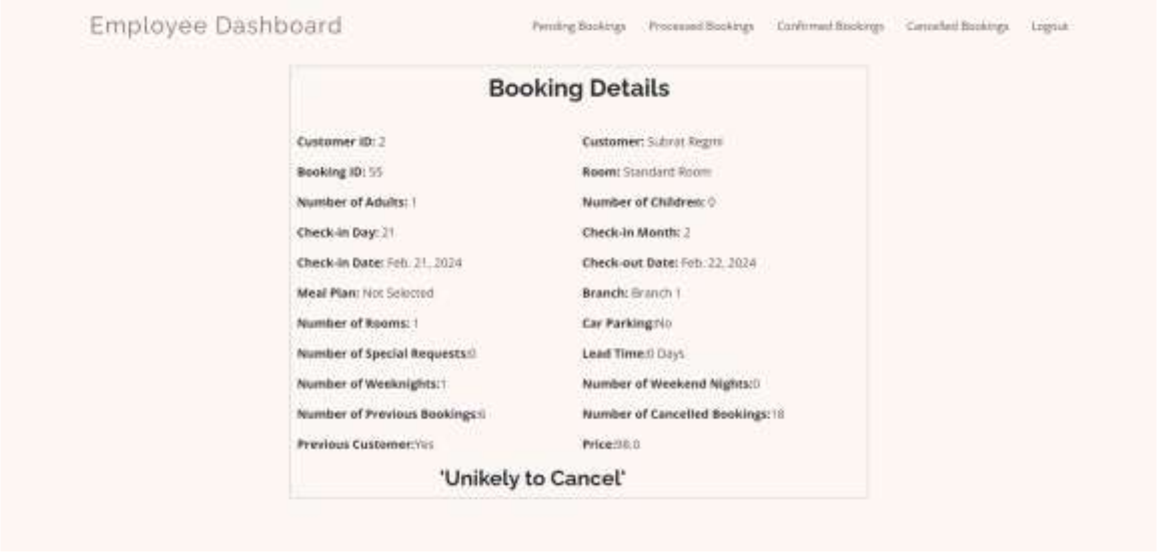
Airport pickup:

Wheelchair:

Number of Rooms: 1

Your Total: $

Appendix E: Booking Form



Booking Details

Customer ID: 2	**Customer:** Subrat Regmi
Booking ID: 55	**Room:** Standard Room
Number of Adults: 1	**Number of Children:** 0
Check-in Day: 21	**Check-in Month:** 2
Check-in Date: Feb. 21, 2024	**Check-out Date:** Feb. 22, 2024
Meal Plan: Not Selected	**Branch:** Branch 1
Number of Rooms: 1	**Car Parking:**No
Number of Special Requests:0	**Lead Time:**0 Days
Number of Weeknights:1	**Number of Weekend Nights:**0
Number of Previous Bookings:6	**Number of Cancelled Bookings:**18
Previous Customer: Yes	**Price:**98.0

'Unikely to Cancel'

Appendix F: Prediction Page