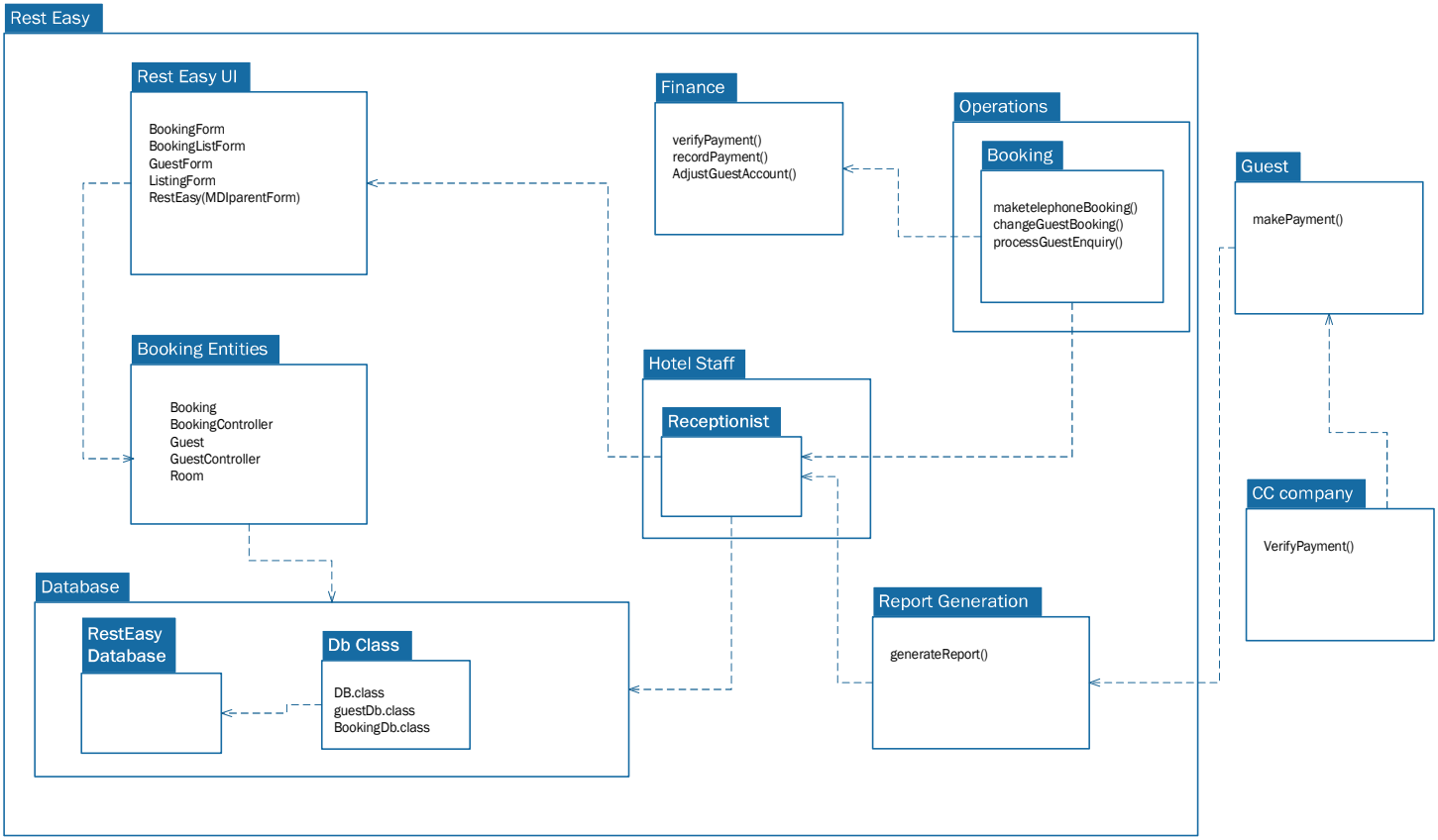
THIS DOCUMENT STATES HOW THE APPLICATION IS TO BE USED AS WELL AS HOW IT WAS MADE. IT ALSO INCLUDES SOME DIAGRAMS AND PROCESSES NEEDED DURING PROJECT MANAGEMENT

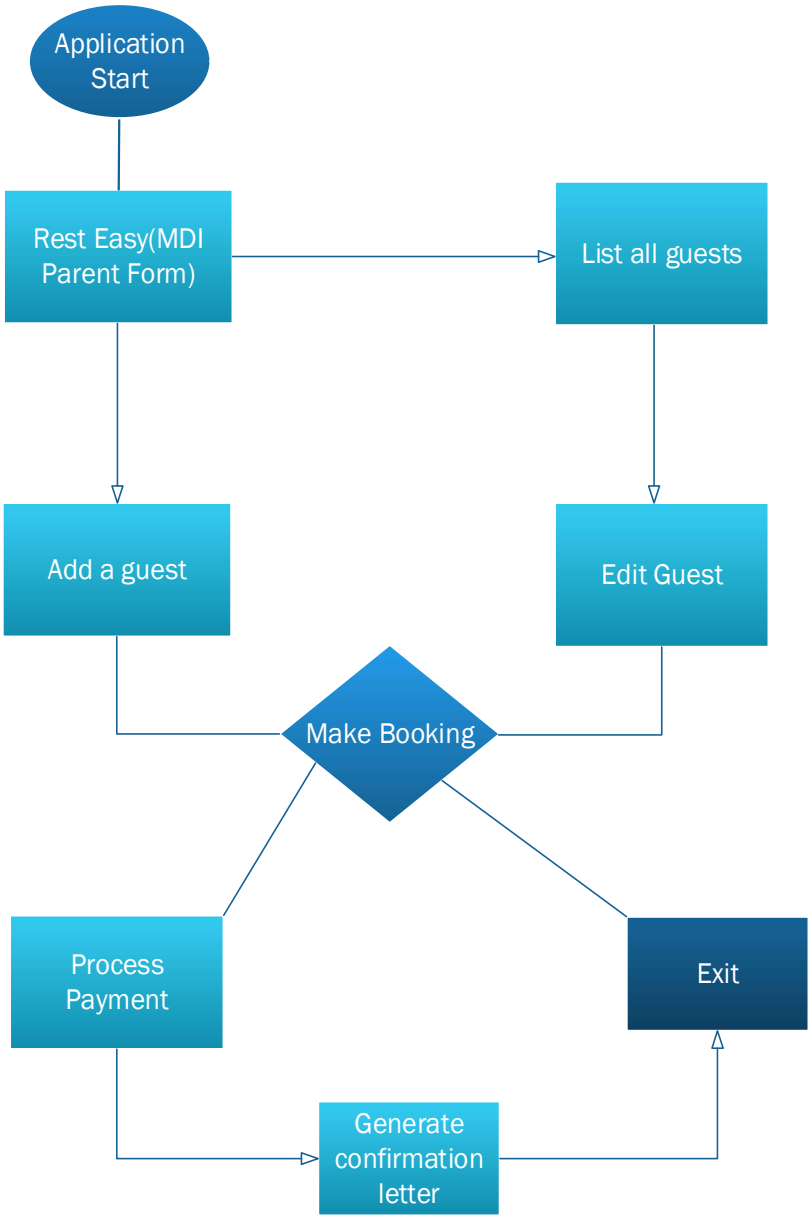
CONTEXT & SCOPE OF SYSTEM SPECIFICATION-PACKAGE DIAGRAM



.

USER INTERFACE & DIALOGUE DESIGN

INTERFACE FLOW DIAGRAM



When the application starts, it starts off by opening the Rest Easy form which is the MDI parent form. The toolstrip menus allow listing of all guests. The Receptionist/user can therefore list all guests when attending to an enquiry by a guest and edit a certain guest or booking. Alternatively, the user add a guest or booking. All two case both lead to making a booking if the guest wants to make a booking. The payment is processed after making a booking through the CC company which confirms the payment and receptionist generates a confirmation letter which will be sent to the guest.

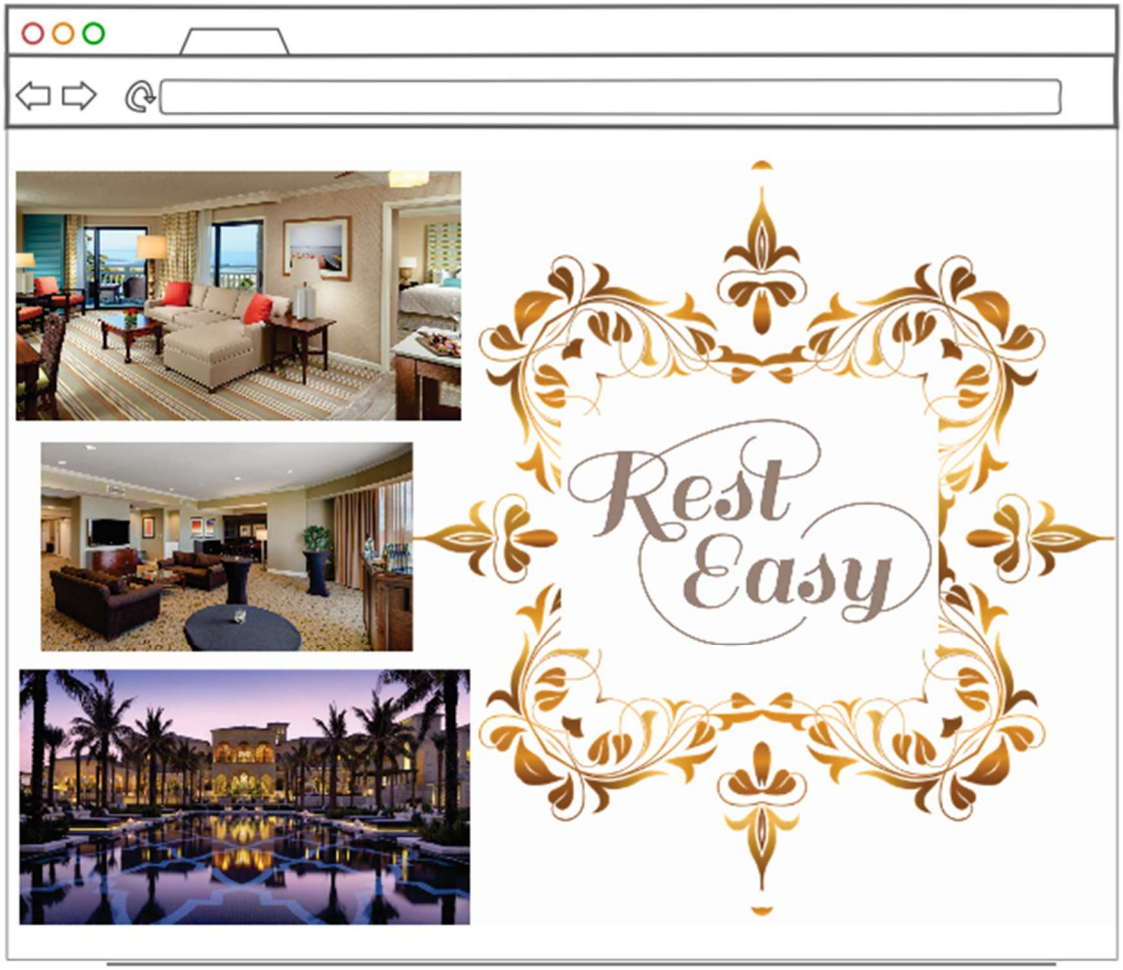
DETAILED SCREEN LAYOUT

Input Data Screens

The start page is the MDIparent form which links to the other child forms that perform the booking operation. The MDIparent form also has basic information about the hotel e.g the contact details and location of the Headquarters of Rest Easy.

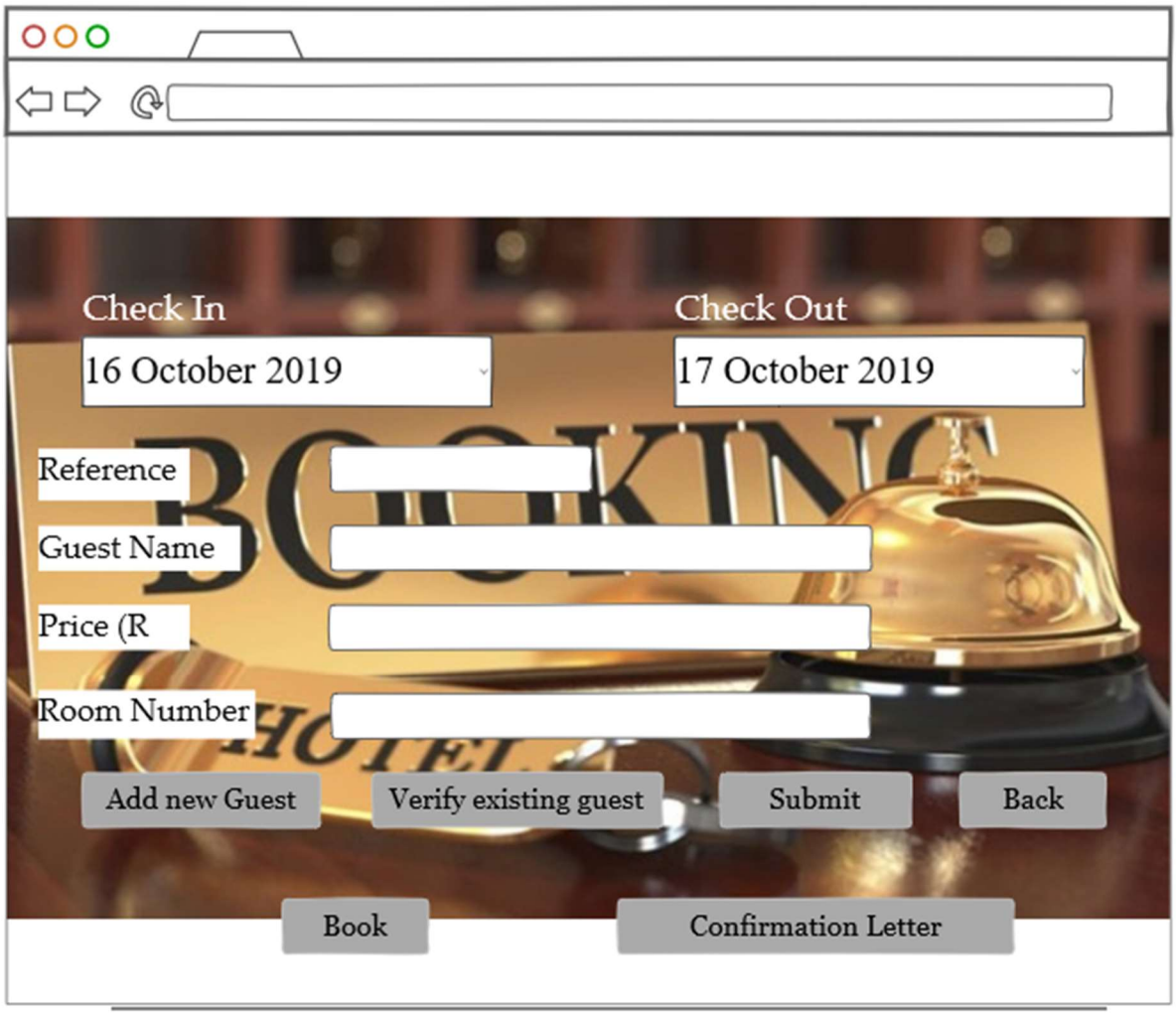
After starting the application, the user can add a guest, delete a guest and edit an existing guest. The MDIparent has the tabs which links access to its child forms namely, booking form, guest form and the listing form.

The (MDIParent form) coded as Rest Easy form



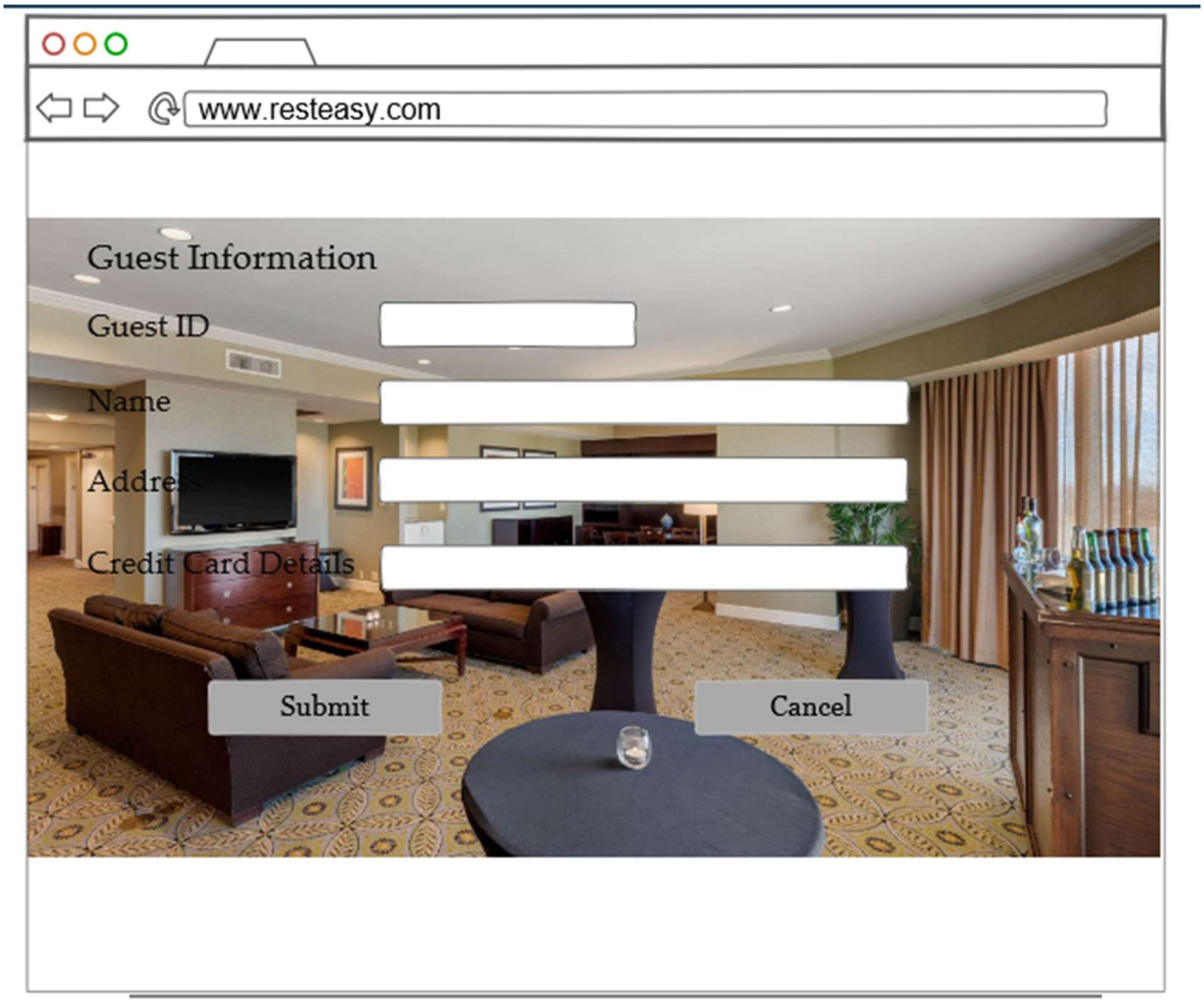
The booking form

The form allows the user to make a booking. The user enters the check In date and the check out date. The check in date and check out date are chosen from the date time picker calendar. The form also verifies an existing guest . The submit button allows the guest to be added to the database as well as the list.



The Guest Form

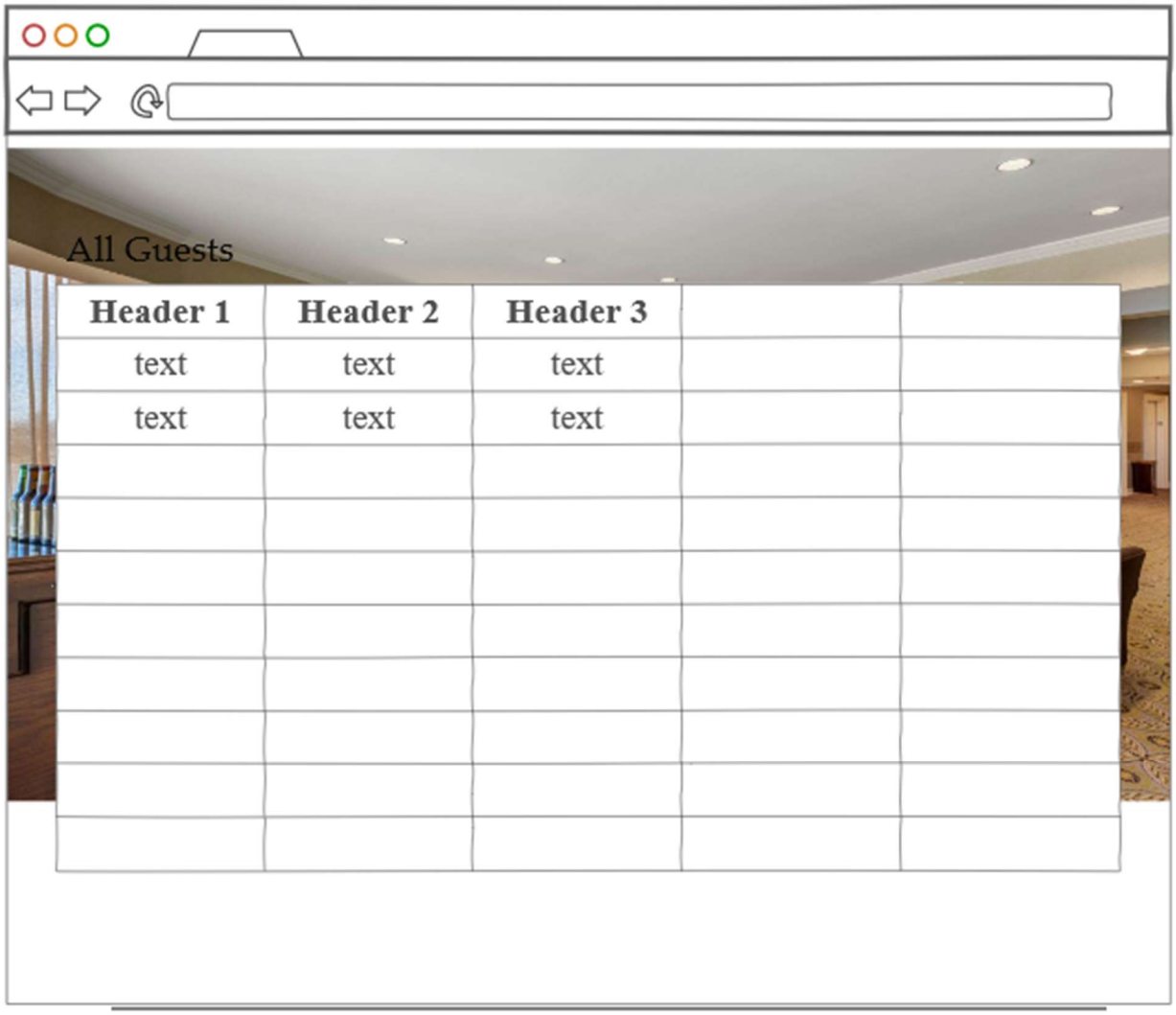
This is where all the guest's personal information is entered. The entries are all keyed values. The guest id is entered in as an integer and does not allow for duplicates as it is unique to each guest. The user then enters the name, address and credit card details of the guest to which the booking is being made for. The values are keyed in as strings.



Besides adding a guest in this form, the user can also edit a guest but cannot edit the guest id as it is only unique to that guest and cannot be changed.

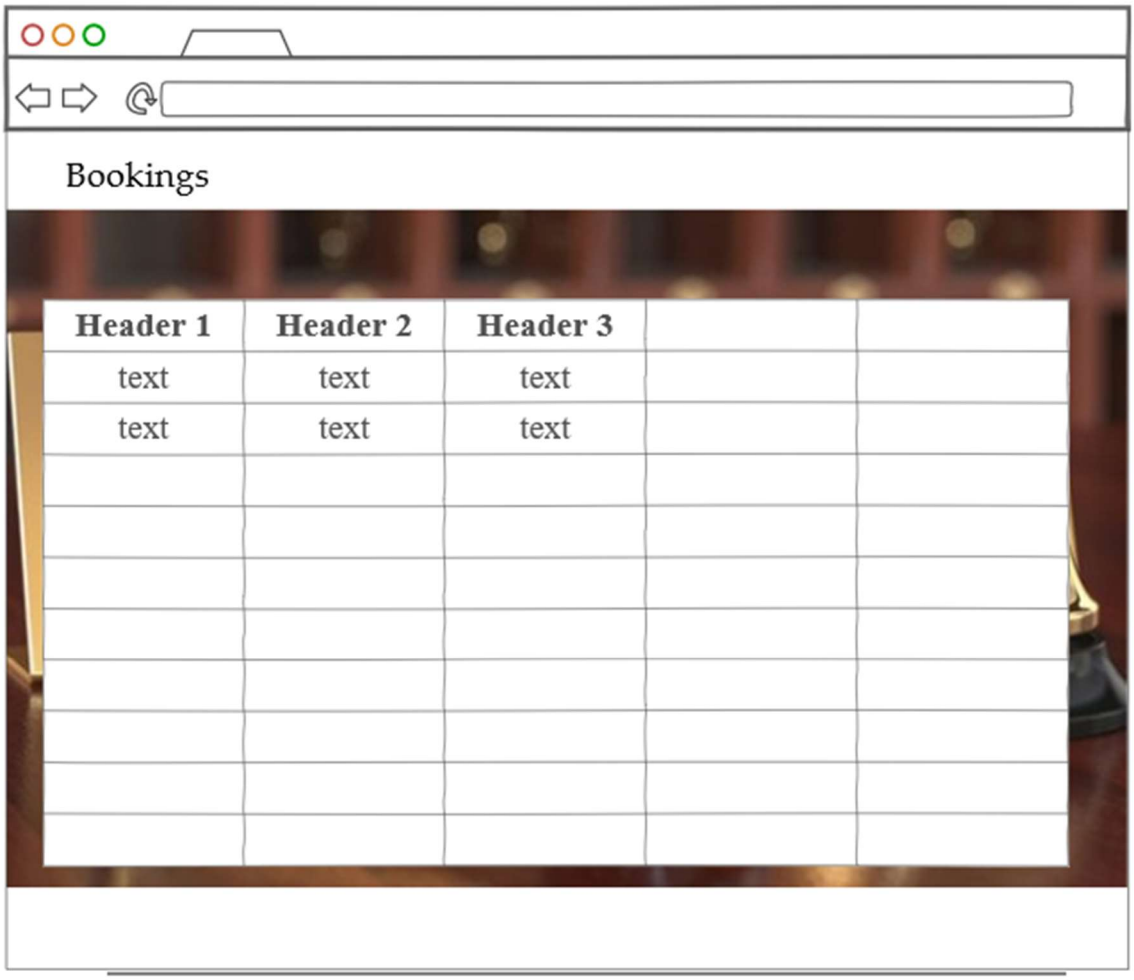
The Listing Form

The form lists all the guests that are in the database. When the user adds a guest, the new guest information should appear as a new row in the guest list view.



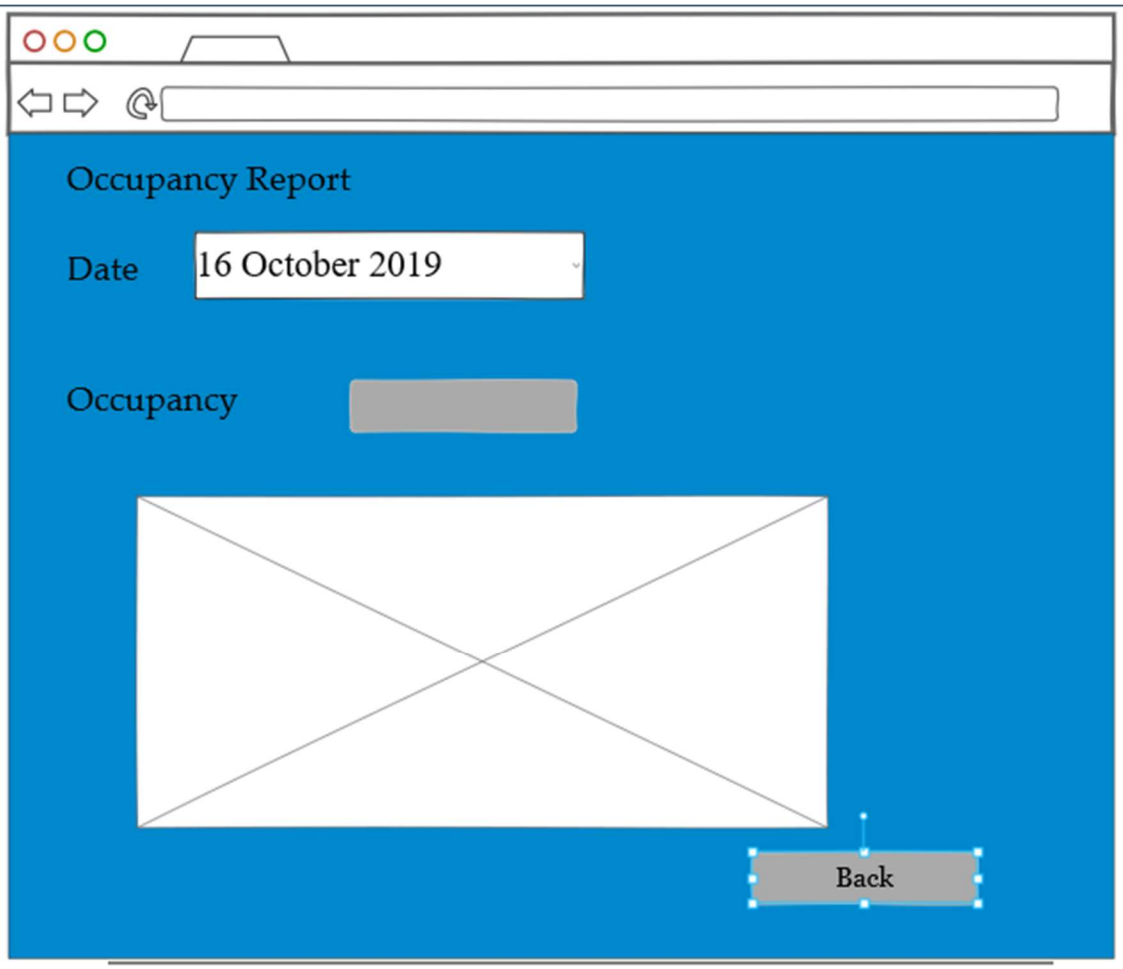
The Booking List Form

The form lists all bookings that are currently in the database.



The report form

The report form shows the occupancy levels determined by the number of rooms occupied on a particular date selected by the user.

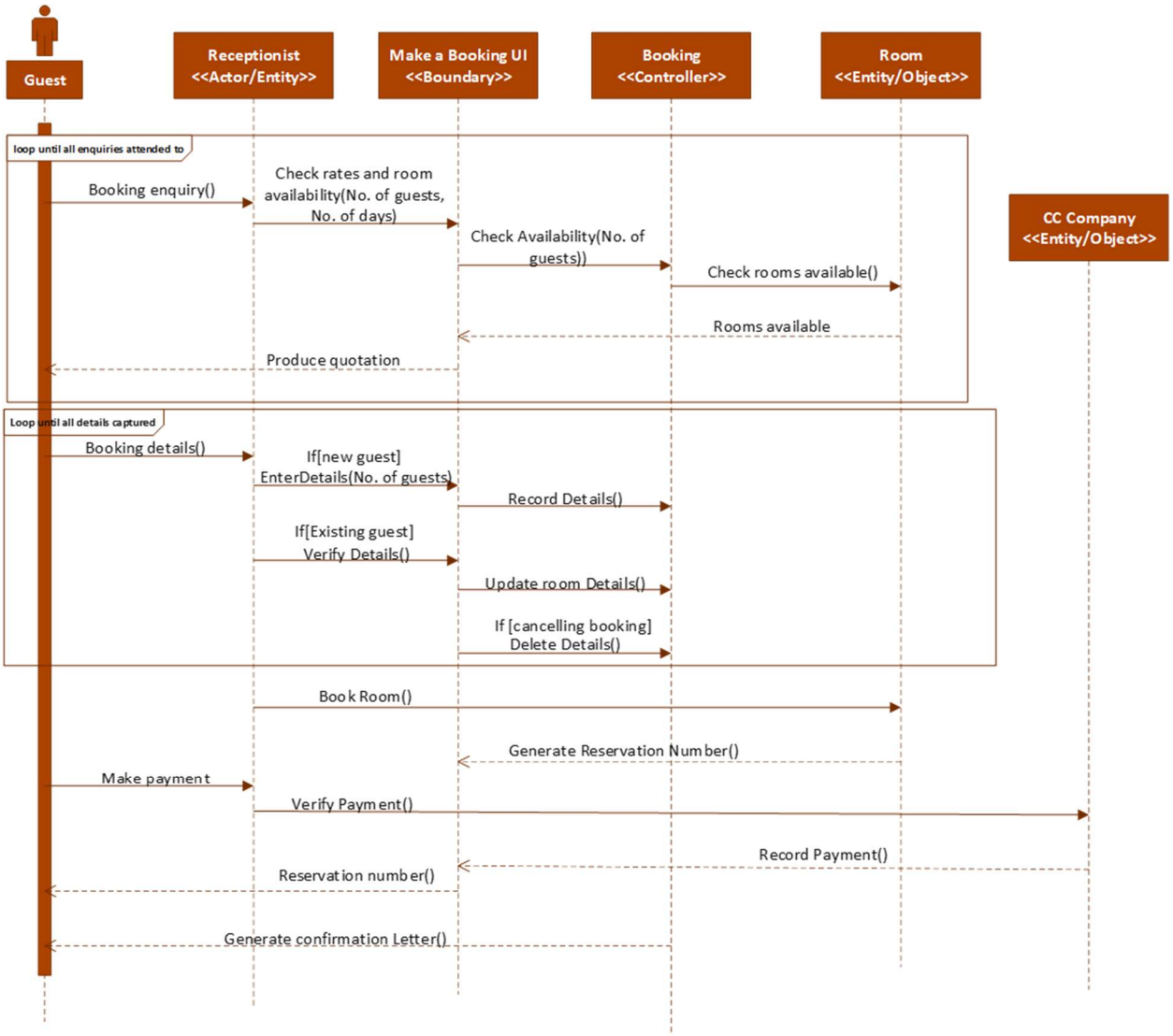


DESIGN SEQUENCE DIAGRAMS

The sequence diagram will model the interaction between objects and/ or actors within the Rest Easy system. The sequence diagram assist in documentation and determining the requirement of the proposed booking system.

DESIGN SEQUENCE DIAGRAM 1

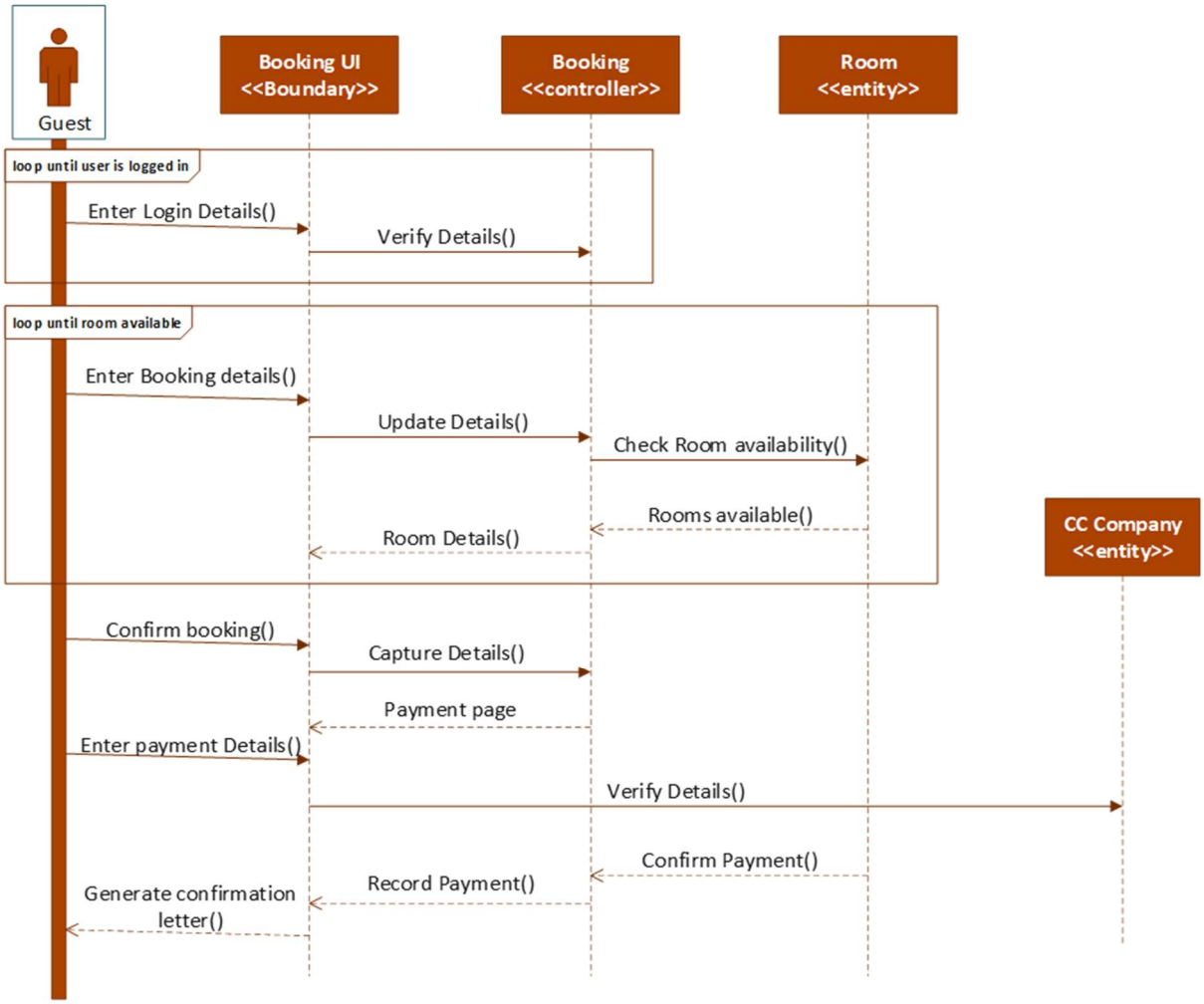
Make a telephone booking sequence diagram.



The diagram models the interaction of the receptionist upon receiving a call from a potential guest/customer.

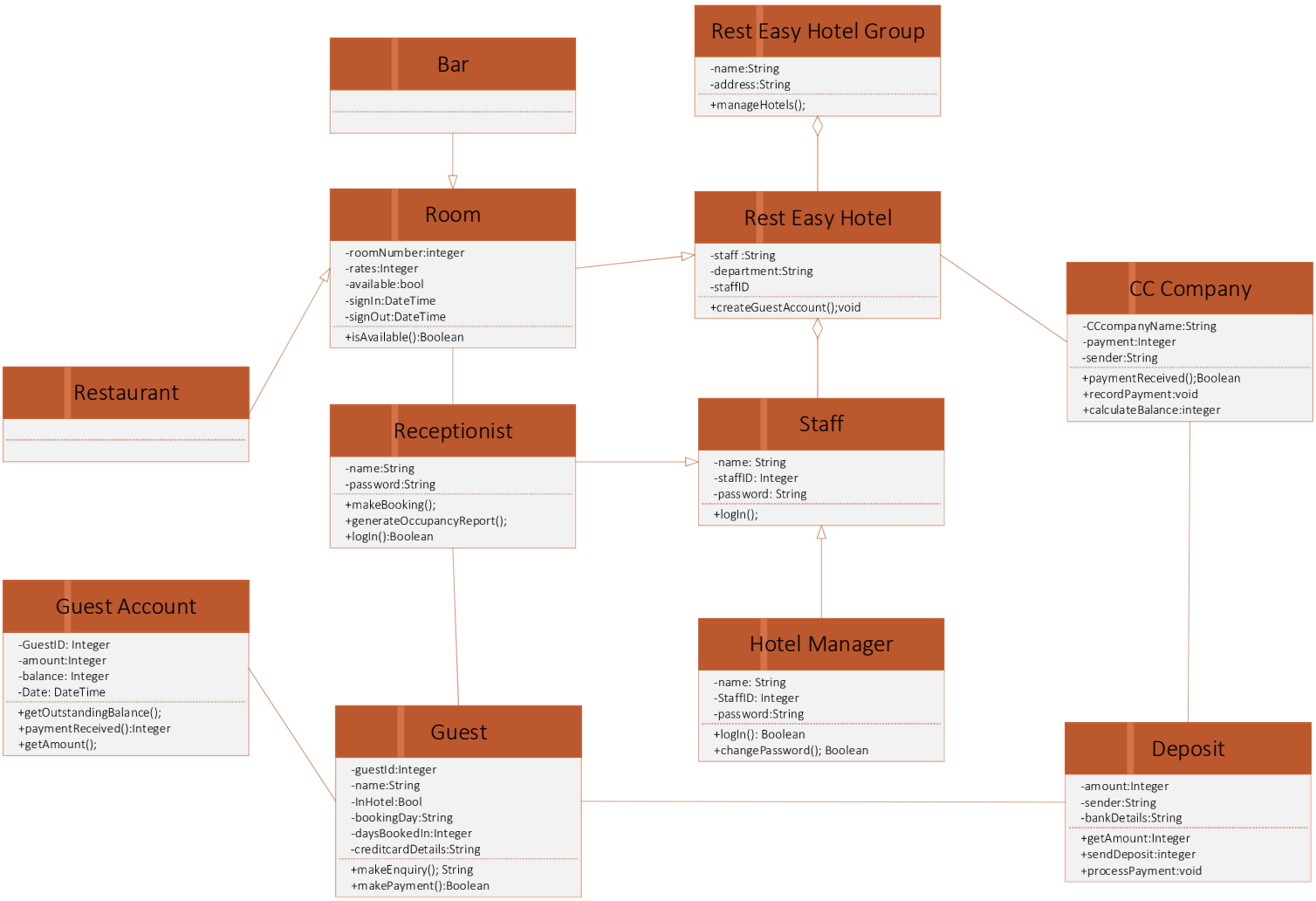
DESIGN SEQUENCE DIAGRAM 2

Make an online booking sequence diagram



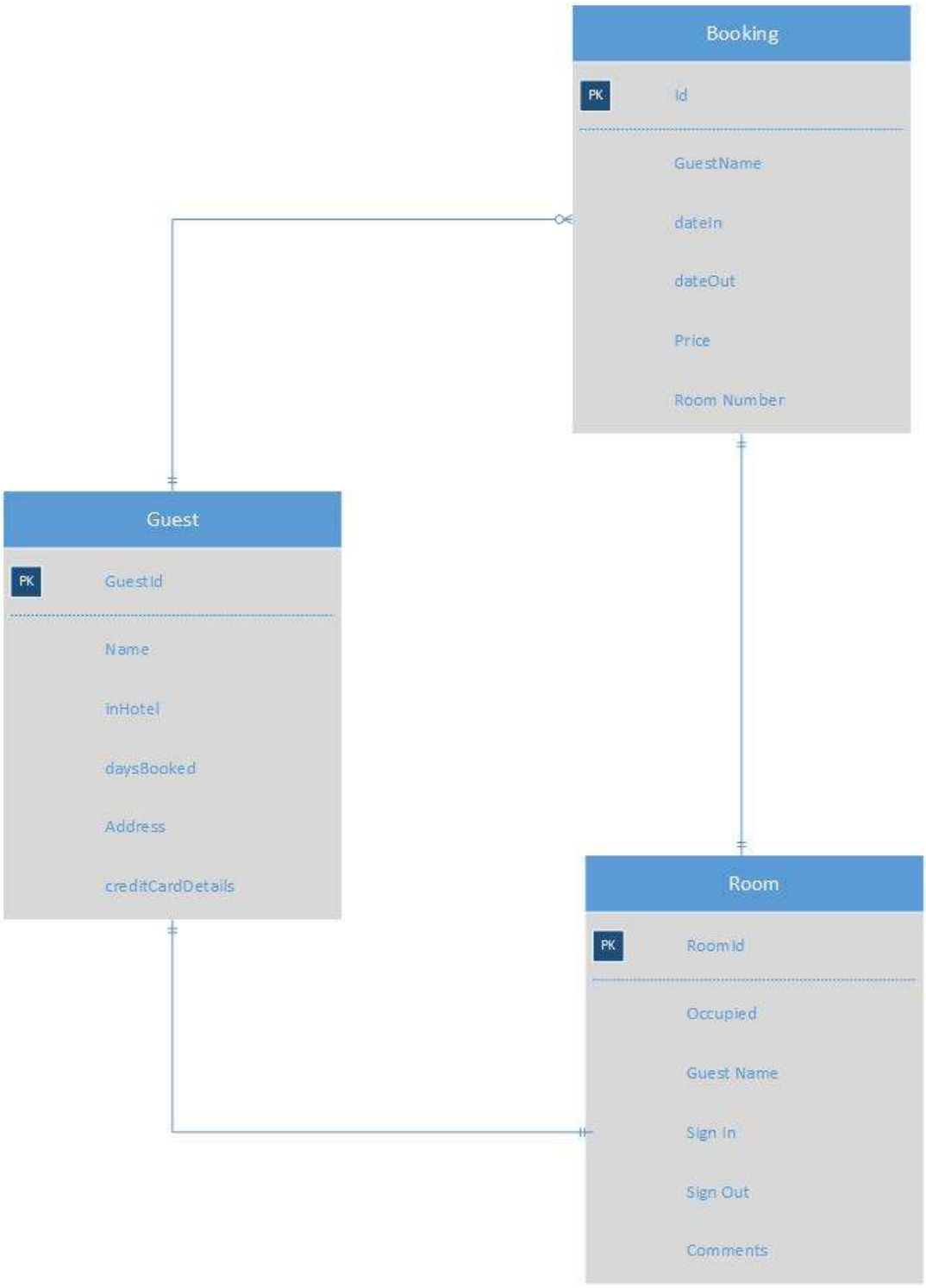
The diagram models the interaction between the guest and the online booking system for Rest Easy.

DESIGN CLASS DIAGRAMS



The Rest Easy hotel group is the master of all Hotels. A single Rest Easy hotel is therefore part of Rest Easy Hotel group therefore the aggregation aspect. Rest Easy hotel processes its payments and deposits through the CC company. The Guest pays the deposit and has a guest account which shows the outstanding balance once the guest checks out of the hotel. The guest directly makes contact with the receptionist for various activities like making enquiries, booking, checking in and checking out. The receptionist is responsible for the booking and therefore has access to available rooms and all the data that pertains to making a booking.

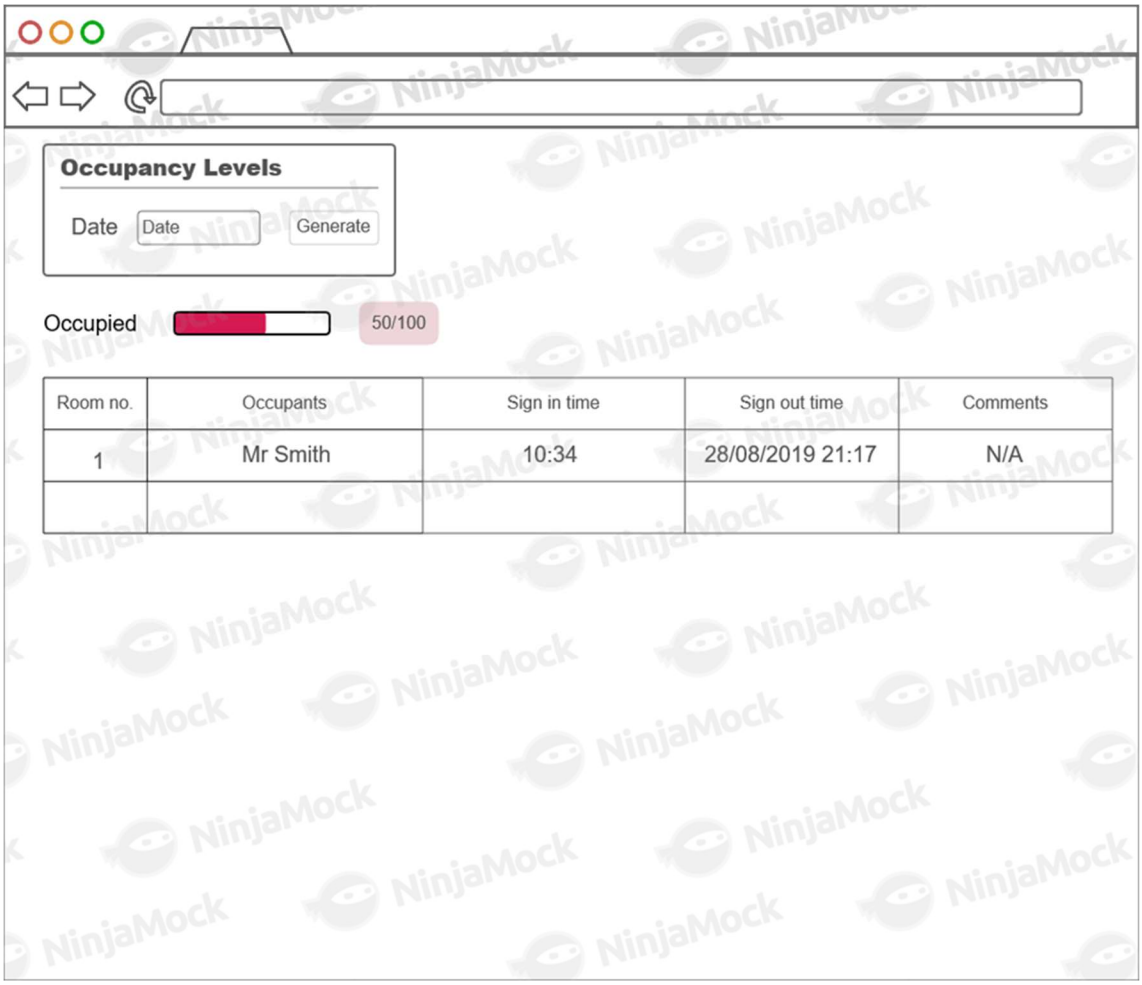
ENTITY RELATIONSHIP DIAGRAM



REPORTS

THE OCCUPANCY LEVEL REPORT

The report will be generated by the receptionist on the day that he/she picks. It is an electronic report



and forms part of the booking application. The report describes the occupancy levels of the hotel. The occupancy level is determined by the number of guests currently booked or staying in the hotel at a certain period of time. Information about the occupancy levels is essential in decision making and strategic thinking when it comes to budgeting and hotel costs.

FORMALISED OUTPUTS:

The confirmation letter displays on the screen after the button confirmation letter is clicked which generates the letter .The button also generates a confirmation letter as a text file. The confirmation will be emailed to the guest as proof of reservation. (The confirmation letter can be found in the folder path

bin>debug>confirmation letter.txt.) The form on text file and screen has all the information the guest needs pertaining to the reservation i.e guestID,room number and so on.

Other formalized forms of output include message boxes that help the user to keep track of what they are doing. For instance, after adding a guest a message box shows to let the user know that the guest has been added in successfully.

TEST PLAN

The test plan aims to test the booking function and detect errors and weaknesses in the system. It tests to see if the user/ receptionist can add, edit, view and delete guests. It also checks for other functionalities like the availability of a room. Generally, all test cases aim to fulfil the telephone booking use case and also taking into account adding, viewing , editing and deleting existing guests or information.

TEST ENVIRONMENT

The test plan requires a computer, fully installed and working visual studio, internet, browsers i.e firefox, chrome, etc, a team of individuals to carry out the test plan. All of these requirements are to be set up in time in preparation for the testing procedure.

TEST ITEMS

Item to be Tested	*Test Description*
Adding a guest	Guest should be properly added to the database and show in the list of guests in the system
Editing an existing Guest -Changing a booking	System should allow for changes to be made to an existing guest. Changes should show when the list of guests is viewed
Deleting a Guest – Cancelling a booking	Once a guest confirms cancellation of a booking, the user should be able to delete entries relating to the guest.
Make a guest booking enquiry	Receptionist should be able to walk through the application seamlessly when the guest is on the

	phone. Receptionist should be able to review prices of rooms as well as the rooms available.
Check Room Availability	The user should be able to check which rooms are unoccupied at a certain period of time and allocate a room to a guest.

TEST APPROACHES

Unit Testing: Each code for a certain functionality e.g code for adding a guest/booking is properly tested for errors after completion. This enhance accuracy as each piece of code is thoroughly tested for errors.

System testing: Done after unit testing. The tests will test the system as a whole to see if it achieves its required functionality.

Beta Testing: the tests will be conducted in a setup that mimics a hotel environment. All the required hardware and software tools will be made available specifically for the test.

Alpha Testing: follows Beta testing. Data that mimics or represents the true actual data is used for the tests. This is done in order to determine errors through using actual data

The booking system will go through development testing and then smoke testing which tests the crucial parts of the system and ensures that the application does not crash when running.

PROBLEM TRACKING (TEST CASES)

The code is first tested using unit testing which tests a piece of code for errors as well as to see if it works as planned. Testing pieces of code makes it easier to detect errors in the system. By breaking down the system into smaller pieces of code allows faster detection of errors and makes the work much lighter and easier. When all the code has been tested and coding of the application is complete, the application will be tested as a whole to check for functionality and flow. After system testing alpha testing will be implemented to test the application with dummy data. Beta testing will then follow after alpha testing. The application will be launched in a virtual hotel environment. After the application passes all tests and is error-free, it will then be launched or implemented.

Example of Test Case:

Unit Testing: Testing the code for adding a booking

Expected output: Booking is added to the system and shows in the booking list

TEST SCHEDULE

Milestone	Duration(days)	Start Date	End Date	Person conducting
Unit Testing	5	15/09/2019	20/09/2019	Software Team/developers
Beta Testing	2	21/09/2019	29/09/2019	Software Team/developers
Alpha	2	24/09/2019	26/09/2019	Software Team/developers, Rest Easy staff
System Testing	2	26/09/2019	28/09/2019	Software Team/developers, Rest Easy Staff
User Acceptance Test	1	28/09/2019	29/09/2019	Project Manager, Rest Easy Staff and customers