



SYSTEMS DESIGN DOCUMENT REST EASY



Akhil Boddu & Jordy Chetty
UNIVERSITY OF CAPE TOWN

CONTENTS

1. INTRODUCTION	3
1.1. OVERVIEW OF SPECIFICATION	3
1.2. CONTEXT & SCOPE OF SYSTEM SPECIFICATION	3
Scope	4
Package Diagram	6
1.3. DESIGN ASSUMPTIONS & CONSTRAINTS	7
Assumptions	7
2. USER INTERFACE & DIALOGUE DESIGN	8
2.1. INTERFACE FLOW DIAGRAMS	8
2.2. SCREEN STANDARDS	9
2.3. DETAILED SCREEN LAYOUT	10
Home	10
add a reservation	11
guest form	12
payment	12
EDIT A RESERVATION	13
make an enquiry	13
3. DESIGN SEQUENCE DIAGRAMS	15
3.1. DESIGN SEQUENCE DIAGRAM - MAKING A BOOKING	15
3.2. DESIGN SEQUENCE DIAGRAM - DYNAMIC REPORT GENERATION	15
Additional Assumptions:	16
4. DESIGN CLASS DIAGRAMS	18
5. ENTITY RELATIONSHIP DIAGRAM	20
ER Diagram Data Dictionary	21
6. REPORT DESIGN	23
6.1. OCCUPANCY Report	23
6.1.1. Detailed Output Requirements	23
6.2 DETAILED Report	26
6.2.1. DETAILED Output Requirements	26
6.2.2. REPORT Layout	27
7. INPUT-OUTPUT STANDARDS & CONTROLS	28
7.1. FORMALISED OUTPUTS	28
7.2. BUILT-IN VALIDATION TO ENSURE REQUIREMENTS ARE MET	28
8. IMPLEMENTATION PLAN	30
MILESTONE 1	Planning, Specifications, Design
30	
Deliverable: Signed off documents and legal agreements.	30

Expected completion and budget	31
MILESTONE 2 Development, Testing	31
Drawdown Conditions:.....	32
MILESTONE 3 Certification, Demonstration, Documentation	32
Deliverable: Working application ready for roll out and all agreements in place.	32
Drawdown Conditions:.....	32
9. TEST PLAN	34
9.1. TEST ENVIRONMENT	34
MINIMUM SOFTWARE AND HARDWARE REQUIREMENTS	34
9.2. TEST ITEMS	35
9.3. TEST APPROACHES.....	36
First testing implementation to be done before development and revised after beta build completed.....	36
9.4. PROBLEM TRACKING (TEST CASES)	36
9.5. TEST SCHEDULE	37

1. INTRODUCTION

1.1. OVERVIEW OF SPECIFICATION

The following document describes the final design of the RestEasy PMS (Property Management System). It follows from the initial proposal on 18 May 2017, and the user requirement specification on 21 June 2017. For the first iteration, the document will focus on the reservation and billing system for Rest Easy.

Each hotel in the group currently has different technology standards and infrastructure which results in inconsistent processes and operations. This prompted the creation of the PMS system to be developed to better manage the operation of the group.

The system is currently in the first iteration of development (beta). Changes were made to the server-side technologies and infrastructure integration technicalities are in the process of being handled. Testing on developed artefacts is to be done afterwards.

RestEasy has appointed Mantaray, a Johannesburg-based company, to review the implementation of the core components of the system and give suggestions on improving the user experience of the system. The system may be changed based on their input.

The following sections in this document describe the technical aspects of the system.

1.2. CONTEXT & SCOPE OF SYSTEM SPECIFICATION

One of the group's important strategic goals is to provide a consistent positive interaction and experience with customers across all hotels. The group wants to improve its brand awareness through the experience they provide to customers which is not possible with the current system.

The software standards in each hotel is inconsistent which makes information and data difficult to attain and manage. The executive management is not able to make effective strategic decisions as a result of this. Some hotels may not have the technical resources to be able to serve clients.

The quality of customer service varies greatly from one hotel to the next. The staff are sometimes unable to perform functions and operations are manual in some hotels which leads to data integrity issues. The security in the hotels is a major concern as the system allows for theft to take place. This has caused the group to fall into disrepute in the past.

The RestEasy PMS system was built to improve on the quality of service and infrastructure offered by the RestEasy Group by standardising the protocols across all hotels.

The implementation of the system will reduce unnecessary running costs, improve consistency across different hotels which is common in the current system, reduce the number of technical-errors that occur in the current system, and prevent theft from taking place.

The scope and the package diagrams are shown below.

SCOPE

In Scope	
Subsystem	Functionality
Front office	
Reservations	Handle all customer reservations and ensure that the booking are promised
Guest Account Management	Record all payments, purchases and tariffs that a customer makes during their stay and add it to their bill on checkout
Room Allocation	Dynamically allocate rooms to guests and update room availability on the system
Enquiries	Provide users with information regarding to a specific query, provide a help function for frequently asked questions, and contact information for different operations within each hotel
Back Office	
Marketing	Interacts with the Group Marketing function to notify each hotel of marketing strategies that must be implemented
Planning and Forecasting	A set of functions and tools that let the staff develop a series of charts and tables based on financial information
Human Resources	
Payroll	Automatically generate the monies owed to each employee and update financial statements
Shift Planning	Automatically generate shifts for each worker, log shift start and end times for each employee and record the number of hours worked
Leave	Interact with Shift Planning and calculate the leave each worker has earned by the number of hours worked
Tax	Automatically generate tax reports based on the company's financial statements for the current or previous year
Training	Allow for management to send notifications of staff training and to manage the training of each employee

Finance and Accounting	
General Ledger	House all financial transactions
Reporting	Generate financial reports based on captured data
Debtors	Manage debtor accounts
Creditors	Manage creditor accounts
Head Office	
Customer Relations Management	Manage customer administrative issues and analyze guest data to improve the RestEasy experience
Training Program	Organise and schedule training for employees across all hotels in the group and generate reports and analytics of training levels of employees in each hotel and the group
Group Level Human Resources	Organise staffing, development, compensation, safety and health, and employee and labor relations
Group Level Accounting	Summarize all business financials into a main console that management can easily interact with
Customer Loyalty Program	Allow management to update the loyalty program as they see fit and coordinate activities as well as easily add and remove guest incentives to the program.
Group Marketing	Automate the collection of guest data and determine usage patterns to aid in sales and management decision-making
Out of Scope	
Credit Card Bank	
Credit Card Authorization	Allow booking payments to be made

FIGURE 1: Scope definition¹

¹ This Systems Design Document outlines the design of the final goal, however for the purposes of this course the first iteration will focus on the “Front office” part of the scope. The assumption is that in the upcoming iterations the above headings that are in scope will be complete.

PACKAGE DIAGRAM

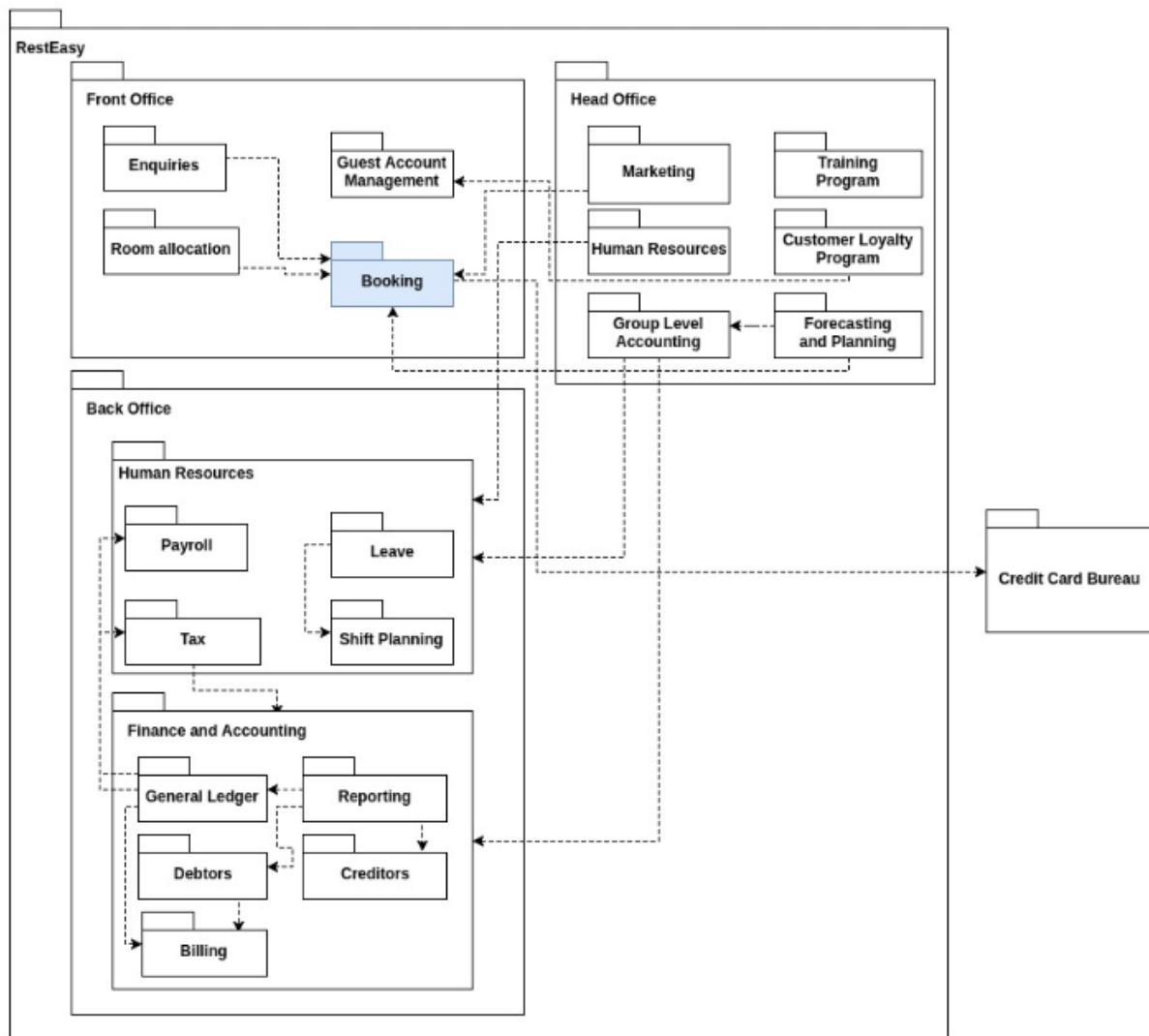


FIGURE 2: Package diagram describing scope visually and highlighting most important aspect of this document (Booking).²

² As seen above, the main focus of the first iteration has been highlighted in blue: Booking. The main aspects that will be addressed as required by the course are Enquiries, Guest Account Management, and the Room allocation process. Once again, the assumption is that, all other parts of the scope will be fulfilled in the later iterations and this Document outlines the scope that is expected to be produced at the end.

1.3. DESIGN ASSUMPTIONS & CONSTRAINTS

Trade-offs often have to be dealt with when building a large system. This section will outline the specifics of the trade-off analysis that will significantly determine the usage and worth of the system in the future.

The system (will also consist of website access in later iterations) needs to be as *efficient* as possible since it will be in use by many staff members and (possibly) customers at one time. Additionally, the system has to be *functional* so that the staff can perform operations that they require for their jobs easily. Such as performing a booking, performing a reorder of stock, the generation of reports and of course communication with guests upon arrival, departure and pre arrival to the hotel. This may however prove to be a conflict since it may be difficult to have an extremely efficient system and have a high functionality at the same time since more time will be spent understanding the functionality hence reducing efficiency. However if the RestEasy team is trained well before implementing the system, they will be able to work efficiently in using the functionality and hence making the booking process and general stay more pleasant.

The system also needs to be *secure* to protect the assets and the reputation of the company. In addition, there should be no room for fraud when it comes to in hotel accommodation. More importantly, the system needs to be *reliable* so that revenue and reputation amongst customers is not lost. Security and reliability go hand in hand, meaning that if a system is secure, it will definitely be reliable as well.

The system has to also be *flexible* so that it can be progressively updated to prevent building a new system and thus reducing costs, and also *maintainable* so as to reduce running costs and be able to consistently and easily upgrade the system. Trying to make a system flexible to making updates and adding functionality may be challenging if it has to also be easily maintainable. (View point 3 below in assumptions)

ASSUMPTIONS

1. The project budget is not a major concern for the group. They are prepared to make a large economic investment for a system that will be in use for the foreseeable future. If need be, RestEasy will increase the project budget if the board sees fit to ensure that critical system functions are properly functional.
2. Staff will be trained on using the system before deployment to ensure top class and optimal usage of the systems key functionality.
3. Since not much additions will need to be added to the system in terms of the reservation process, maintainability will not be a problem.
4. RestEasy has a good security infrastructure such as key card access to certain rooms and physical access such as data of guests will be secure as well as any physical access to money.
5. This project mainly focuses on the first iteration of development. As the iteration stage is progressed, new diagrams will have to be drawn based on the required functionality of the system.
6. For the report generation, the data will be generated electronically for both the occupation report and the other detailed report.

2. USER INTERFACE & DIALOGUE DESIGN

The user interface is an extremely important part of design. Being able to make users interact with the system with ease will affect the productivity of the workforce when dealing with bookings on the front end and of course the in-house management (back end). This section will provide a graphical representation of the user interface and explain the reasoning behind design decisions. Additionally, the flows between the different screens will be shown through the Interface flow diagrams and finally the justification and application of design principles will be explained.

2.1. INTERFACE FLOW DIAGRAMS

The fundamental reason of having so many interactions is to always provide the user with the option of going back to the previous screen, or simply cancelling the reservation and just checking for availability. Additionally, it makes the design on each screen neater and fully functional since whatever is necessary for each screen is shown. The forms will follow Shneiderman's 8 Golden Rules of Design, and the flow has been strategized and planned based on the design principles.

Below is the interface flow diagram that models the current goal of the system in terms of interactions between screens.

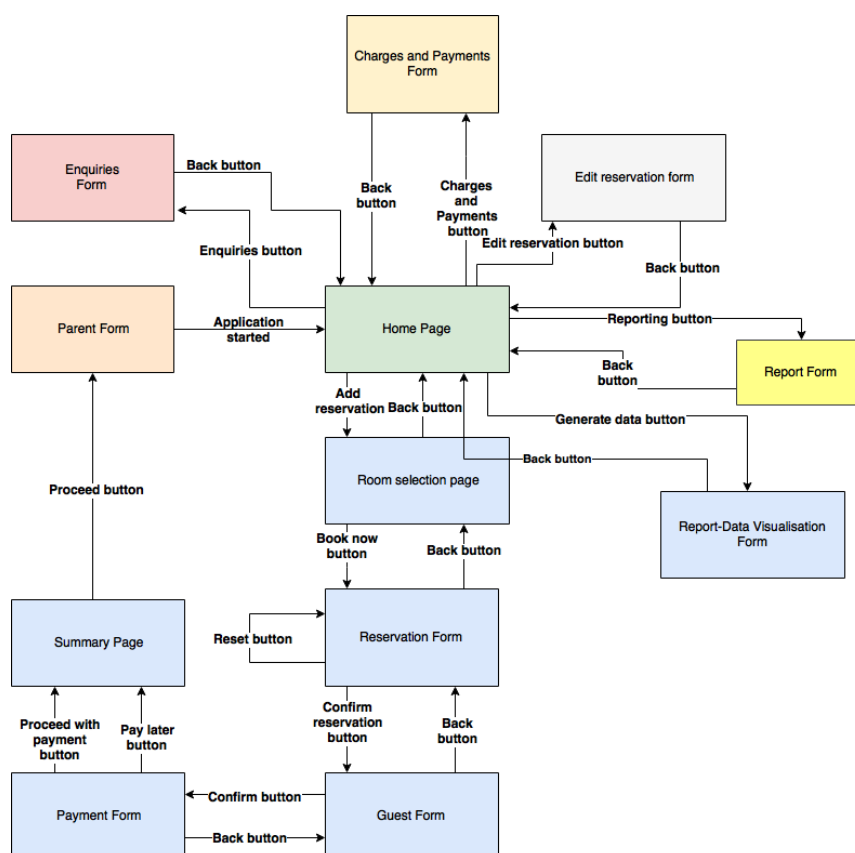


FIGURE 3: Interface flow diagram³

³ Expandable interface flow diagram: <https://goo.gl/b7gTxk>

2.2. SCREEN STANDARDS

Screen standards are vital to ensuring that there is a flow and the system is user friendly. Users can only be extremely productive if the interfaces are user friendly. To make the screens more user friendly they need to be able to follow a certain standard and aesthetic so that users can easily predict what to do even though they have not actually the system before.

Below are the layout, colours and appearance expected from the system.

1	LAYOUT
	<p>The layout follows Shneiderman's 8 Golden Rules of Design:</p> <p>1 Strive for consistency.</p> <p>The whole systems follows the same aesthetic and informal rules. For instance on each screen, the user can navigate accurately since there will be titles and relevant labels to guide the user when necessary.</p> <p>2 Enable frequent users to use shortcuts.</p> <p>As the frequency of use increases, users can navigate through forms faster by using the drop down menu on the toolbar of the MDI Parent. Additionally, users can hit the tab button to navigate through the forms tabs.</p> <p>3 Offer informative feedback.</p> <p>The user is often notified of the action taken with a message box, or a change in color and visually grabbing activities (Form switching, as well as panels appearing and disappearing).</p> <p>4 Design dialog to yield closure.</p> <p>Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.</p> <p>5 Offer simple error handling.</p> <p>As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error. Users are guided with the help of message boxes when actions that defy the rules occur.</p> <p>6 Permit easy reversal of actions.</p>

	<p>This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. Users can easily cancel a reservation midway of the process and select a date for the booking by simple flows. Users can choose to not book at all thanks to providing the freedom to do so.</p> <p>7 Support internal locus of control.</p> <p>The users of the system are in sole control of the database manipulation. Nothing happens in the background. Any action that is performed by the user will have a direct impact on the system as necessary. Hence the user will have full control.</p> <p>8 Reduce short-term memory load.</p> <p>Users will not need to hold any information in their memory. Details that are necessary for making decisions are always provided when needed. For instance if the user wants to make a payment using the details of the reservation, the price that is needed to pay will always appear and stay on the form.</p> <p>Additionally:</p> <p>Buttons and labels to be neatly aligned. Components of the screen to be appropriately grouped according to function and value to the current activity. Common controls clearly displayed and easily accessible. Auxiliary operations should occupy remaining available screen space.</p>
2	COLOURS
	<p>Each screen to be themed in the colours of the group, white, cyan blue and black. The RestEasy logo should be displayed on the top-left of every screen. The design is minimalistic sticking to the basics of colours with black and white for text and buttons, as well as the cyan blue colour for overlays and shades.</p>
3	OVERALL APPEARANCE
	<p>Minimally designed. Unnecessary functions for the current activity it should not be displayed. Easily navigable and intuitive to use. Operations iconized where possible. The application will run extremely concisely will have a simple aesthetic that will perform complex actions. The screens will never be cluttered and only what is necessary will be displayed.</p>

2.3. DETAILED SCREEN LAYOUT

Below are the screens that will be used to model the screens. Details about input screens, security considerations and design decisions will be explained under each screen title as a description.

HOME

Below is the Rest Easy home screen that is loaded along with the MDI form and will direct to all other forms using the images. A tooltip will appear when the user hovers the mouse over the relevant icon. The aesthetic is minimalistic sticking to the basic black, white and cyan colours.

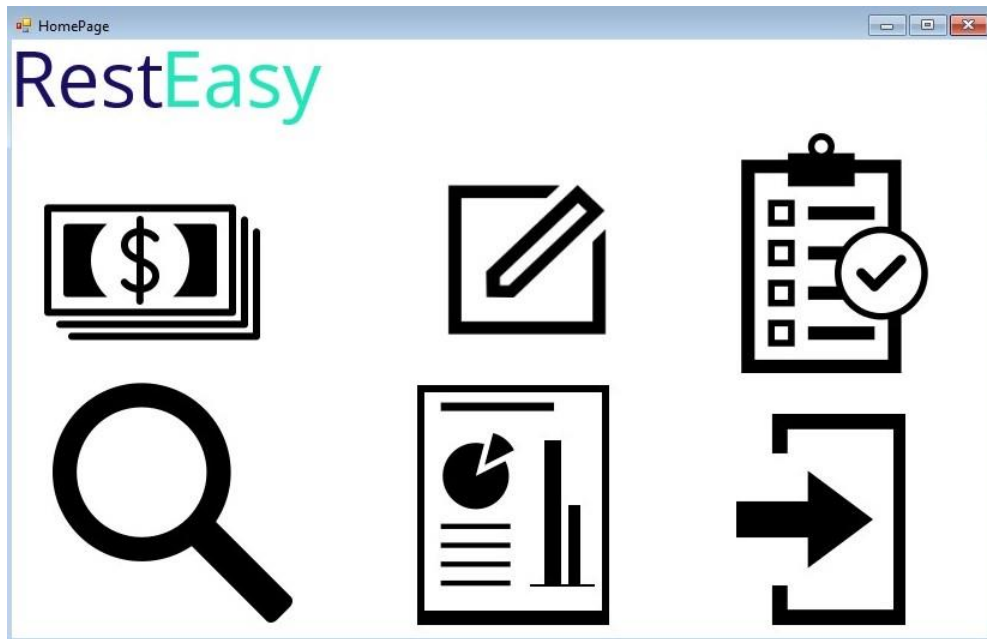


FIGURE 4: Home page

ADD A RESERVATION

Can click on the two buttons, one for selecting the start date and the other for selecting the end date. No input is occurring here, but the user's selection definitely has an impact on the screen. The user cannot click the same button twice to make a booking, since a booking can only be made for at least one night. The main assumption here is that the bookings will only take place in December, but in future iterations this can easily be changed by adding the functionality of selecting a month. The user input is therefore controlled to ensure that bookings are not exploited. This stage of adding a reservation is only reached after the rooms have been selected. For the purpose of this iteration, only one room type is being used and that will be shown as a separate screen.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

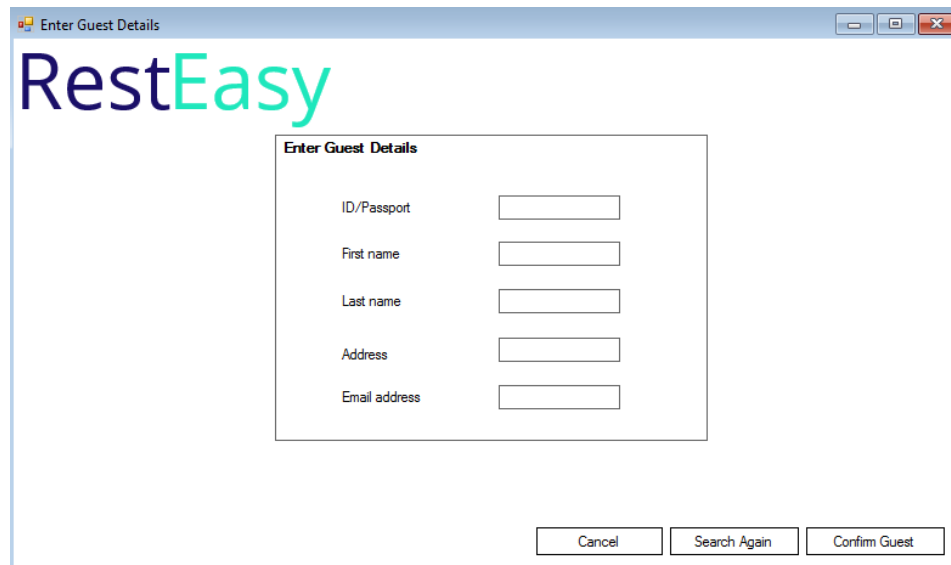
Unbooked
 Partially booked
 Fully booked

Reset Selection Cancel Confirm Reservation

FIGURE 5: Adding a reservation page

GUEST FORM

After the user has completed the date's procedure, the guest details have to be checked. To do that the following form is shown.



The screenshot shows a window titled "Enter Guest Details" with the RestEasy logo in the top left. Inside the window is a smaller box also titled "Enter Guest Details" containing five text input fields: "ID/Passport", "First name", "Last name", "Address", and "Email address". At the bottom of the window are three buttons: "Cancel", "Search Again", and "Confirm Guest".

FIGURE 6: Entering guest details

First name and last name in the above form cannot contain numbers. Additionally, the user can enter the ID/Passport number once and if the details exist in the database the details will automatically appear. All other fields above do not have any security. The email address field can have a dropdown menu to specify the email address provider, however if the email is of an organization then the other option can be specified.

PAYMENT

The input forms below will ensure validation by only taking string and no digits for the Card holder name. Additionally, the credit card number will only take in 13 digits and no more. The CVV Number field will ensure that only digits are entered. To ensure security, the OTP pin will be hidden and asterisks will mask characters.

FIGURE 7: Payment details page

EDIT A RESERVATION

The combo box will restrict the various options to the user to only being able to cancel or make changes to an existing reservation. The start date and end date fields only take in the date types. The date fields cannot be accessed until the ID/Passport details are verified by searching the database to see if it exists.

FIGURE 8: Edit a Reservation

MAKE AN ENQUIRY

Users can make an enquiry about a reservation by viewing the reservation and confirming details. Or simply viewing the flagged guests who are yet to make their deposits. User types in reference number and all details of the reservation are returned to the screen via another panel. Options to make strange enquiries are temporarily controlled by the combobox since the user is a reservation clerk. If the clerk has any admin issues or enquiries it can be addressed within the house. This is purely for the guest's enquiries that will be informed via the reservation clerk.

The screenshot shows a web application window titled "EnquiryForm" with standard window controls (minimize, maximize, close). The main heading is "RestEasy" in a large, stylized font. Below the heading, there is a label "Enquiry Type" and a dropdown menu currently displaying "View a Reservation". A horizontal line separates this header section from the main content area. The main content area contains the instruction "Please enter the Reservation reference number and click on Search" in italics. Below this, there is a label "Reference Number" followed by a text input field. Underneath the input field is a "Search" button. At the bottom right of the form, there is a "Back To Home" button. The entire form is enclosed in a light gray border.

FIGURE 9: Enquiry form

The above screens are the meat of the system and can definitely be polished further. This is the basis that can be built upon to create a more aesthetically, yet fully functionally pleasing system.

3. DESIGN SEQUENCE DIAGRAMS

Creating sequence diagrams are extremely important for modelling certain use cases and visually trying to understand how objects interact and trying to use that model when coding the system from the ground up.

Two sequence diagrams have been modelled for this document, one is the “making a booking” and one is the “dynamic report generation”. Creating the Make a booking is a fundamental functionality of the system and more details on why the modelling of dynamic report generation is vital is mentioned in section 3.2 below.

3.1. DESIGN SEQUENCE DIAGRAM - MAKING A BOOKING

Below is the sequence diagram that visually demonstrates how a room is booked on an object oriented level. (Please zoom in for more clarity or visit <https://goo.gl/xr2vbD>)

Making a booking works by allowing the receptionist to enter the relevant reservation details, they will check for availability and a response will be returned to the receptionist. The receptionist can then enter details of the guest and then the database is checked to see if the guest details are there or not, if the details do not exist the items are added in the database. Once the data exists, a conditional reservation is made where and the condition is for the user to make the payment. Then in the next step payment is made and the user is authorized of the payment.

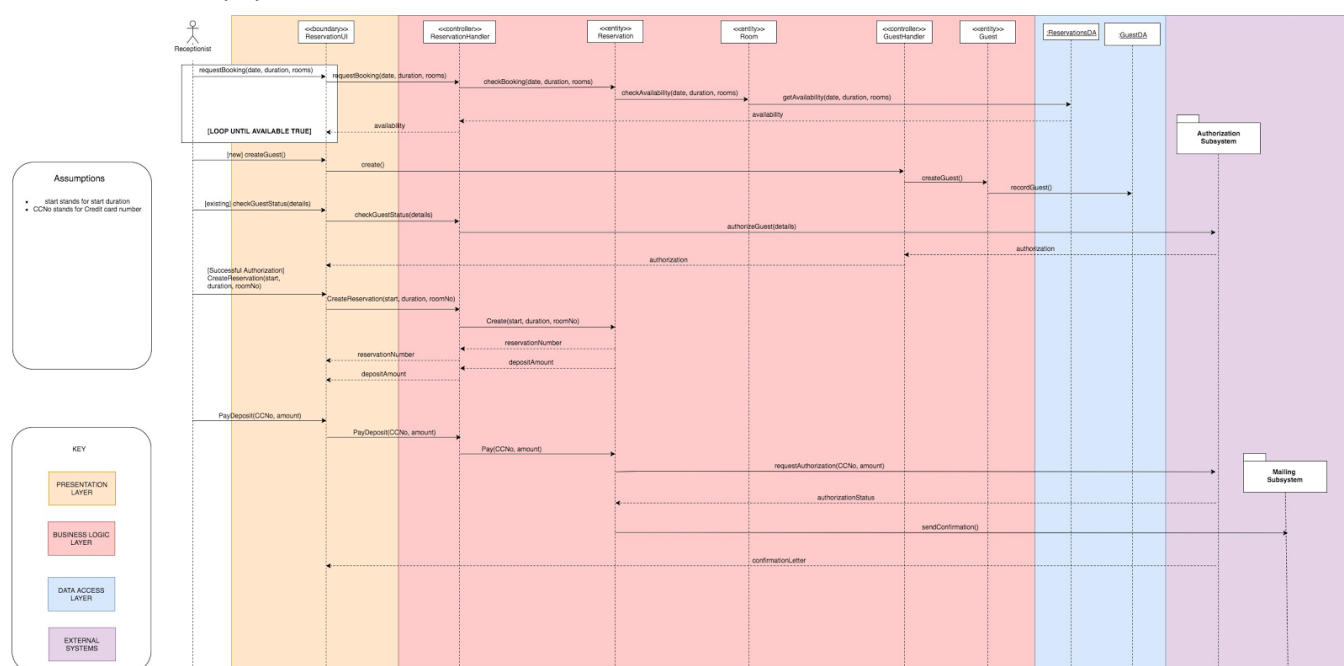


FIGURE 10: Design sequence diagram for making a booking.⁴

3.2. DESIGN SEQUENCE DIAGRAM - DYNAMIC REPORT GENERATION

Most organizations tend to struggle with validation and verification of various details. The solution of the past has been to scavenge through databases and find specific entries to see where errors have

⁴ Link to view Design Sequence diagram for making a booking: <https://goo.gl/xr2vbD>

happened, or use the database to manually figure out trends in the business and try create a report out of the data. To get the aforementioned executed, a database analyst has to be hired, but with the exponentially growing technology and the advances in machine learning and deep learning this tedious process can be nullified.

This project will address this problem by dynamically creating reports, whether it is occupational or financial. These reports will then be compiled and customizable to create a well-defined Summary report or an Executive report.

The receptionist or any other RestEasy official who has the relevant authority to view confidential information can view the produced financial or occupancy reports at this level of development by simply typing in the start data and the kind of report required.

Below is the process of how the data from the report is produced and this is all taking into consideration the analysis of the data. (Please zoom in for more clarity or view diagram on <https://goo.gl/ByZF5q>)

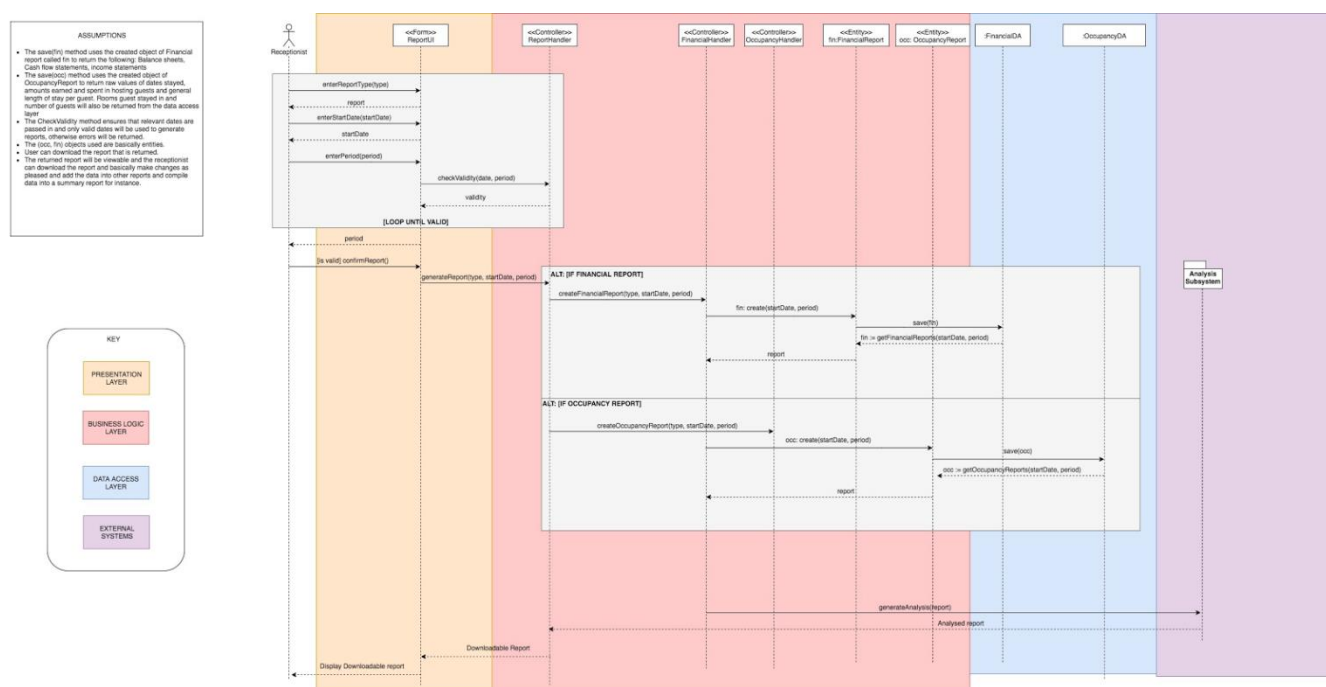


Figure 11: Design sequence diagram for Dynamic report generation.⁵

ADDITIONAL ASSUMPTIONS:

1. Can pass in (PERIOD) time type parameter, to generate monthly, weekly, quarterly, yearly reports.
2. Different kinds of reports can be passed in: Occupancy / Executive

⁵ View Design sequence diagram for Dynamic report generation on <https://goo.gl/ByZF5q>

3. *A validation checked is passed in on the diagram so that valid dated reports are produced*
4. *Need to make sure to get summary of occupancy from occupancy report into the executive report*
5. *Analysis subsystem - external Data analysis API that analyses financial docs to produce graphs, technical diagrams and whatever is necessary for an executive report. There are*
6. *After report is returned, user can make changes to the document - eg layout changes, adding comments*
7. *End date is automatically calculated using the start date and period variables.*
8. *The diagrams that are generated will be compared with the values from the immediate previous period. This will be done by the analysis subsystem.*

4. DESIGN CLASS DIAGRAMS

The Design Class Diagram below heavily focuses on the first iteration of the Project Management System which is the Reservation and Billing System. The diagram below should assist the developers during the development stage. Kindly zoom in or click on <https://goo.gl/Kwrhtp> to view the diagram more closely. Assumptions relating to the diagram are detailed and explained in the box on the top left corner of the diagram.

Diagram on next page

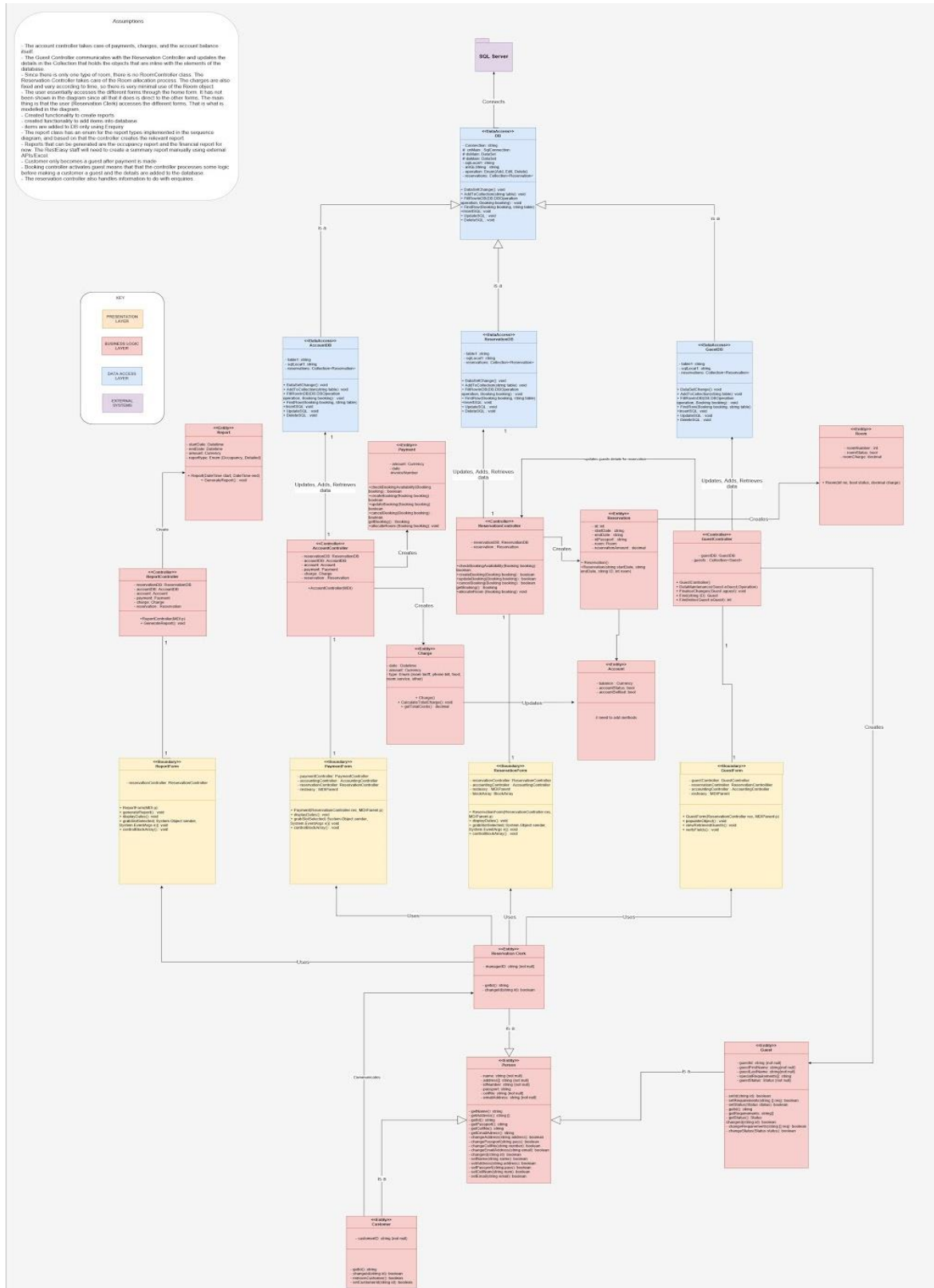


Figure 12: Design Class Diagram.⁶

⁶ Expendable version of The Class Design Diagram on: <https://goo.gl/Kwrhtp>

5. ENTITY RELATIONSHIP DIAGRAM

As in most commercial environments today, the system will be implemented using a relational database. Model and below is the Entity Relationship Diagram using Crows-foot Notation, ensuring all Entities and Associations have correct Cardinality and Multiplicity. The ERD goes beyond simple domain entities, and is fully normalised to 3rd Normal Form (3NF).

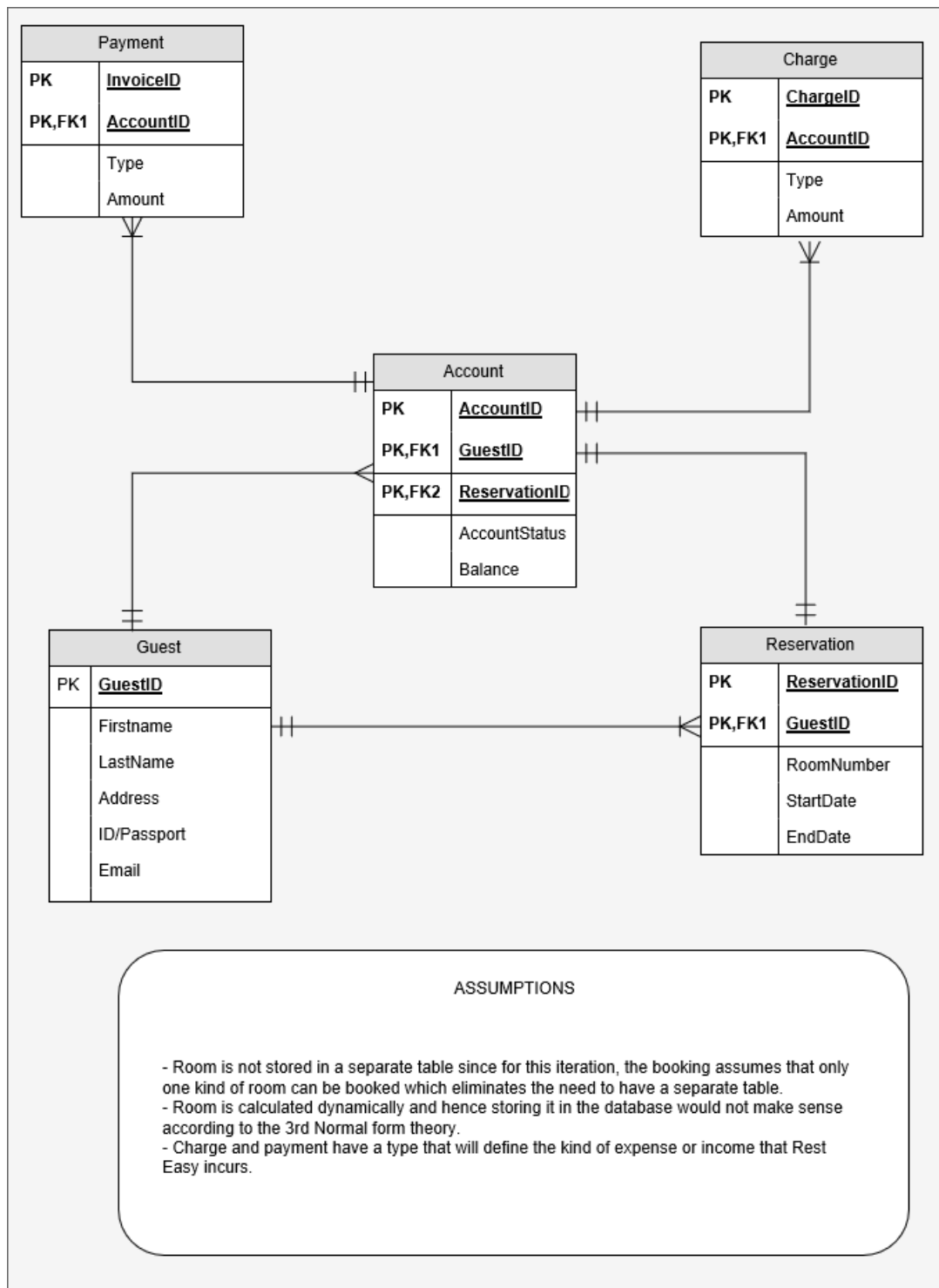


Figure 13: Entity Relationship Diagram.⁷

ER DIAGRAM DATA DICTIONARY

Below is a table describing each entity field of the database exhaustively.

Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
Reservation Entity	Reservations are made by reservation clerks to allocate a room to Guests.					
<u>ReservationID</u>	Unique Identification of a reservation	int	10	true	false	true
<u>GuestID</u>	Foreign key. Unique identification in another table	varchar	20	true	false	true
StartDate	Guest expected arrival date	date	default	true	false	false
EndDate	Guest expected leaving date	date	default	true	false	false
Guest Entity	Guest details are captured if reservations have successfully been made. Guest DB is accessed and retrieves information if guest already exists.					
<u>GuestID</u>	Uniquely Identify a Guest	varchar	10	true	false	true
FirstName	First name of Guest	varchar	20	false	false	false
LastName	Last name of Guest	varchar	20	false	false	false
ID/Passport	Passport or Identification details	nvarchar	20	false	false	false
Email	Guests email address	varchar	100	false	true	false
Charge Entity	Guests incur charges when they are at the hotel. This is noted in this table.					
<u>chargeID</u>	Unique identification of each charge	int	10	true	false	true
<u>AccountID</u>	Account number for Guest Account (Foreign Key)	int	10	true	false	true

⁷ Expendable version of The Class Design Diagram on: <https://goo.gl/Kwrhtp>

balance	Guests current account balance that is due/paid	decimal	default	false	false	false
type	Type of charge that is incurred	varchar	200	false	false	false
Payment Entity	Guests have to pay for the charges they have incurred and that is noted in this table.					
<u>invoiceID</u>	Unique identification of each charge	int	10	true	false	true
<u>AccountID</u>	Account number for Guest Account (Foreign Key)	int	10	true	false	true
balance	Guests current account balance that is due/paid	decimal	default	false	false	false
type	Type of charge that is incurred	varchar	200	false	false	false
Account Entity	All guest charges are added to their hotel account. Each guest has an account that has a balance that is continuously changing and is regularly updated.					
<u>AccountID</u>	Account number for Guest Account(Primary Key)	int	10	true	false	true
<u>GuestID</u>	Foreign Key	varchar	20	true	false	true
<u>ReservationID</u>	Foreign Key	int	10	true	false	true
balance	Guests current account balance that is due/paid	decimal	default	false	false	false
status	Status on whether account is opened or closed	bit	default	false	n/a	n/a

6. REPORT DESIGN

Report design is an extremely vital part of being able to make decisions as an executive. These decisions go on to shape the path a company will take in the future. It is therefore vital to produce reports that will be vital and relevant to the RestEasy executive.

The reports that will be produced are the Summary report and Executive report. The reason both these have been selected is due to the fact that a Summary report gives a summary of performance within the corporate in various departments and that will encourage the company to allocate more resources where necessary in order to make business decisions that have a significant overall impact in terms of generating more revenue. Since RestEasy is such a huge brand, the public and other companies in the industry will expect some sort of executive report to understand how an organization such as RestEasy works to see what decisions are made to expand the business and keep it running successfully. This shows professionalism of the RestEasy branding and creates a standard that other companies in the service sector have to live up to and try to scale accordingly hence providing RestEasy with the opportunity to monopolize the service sector in South Africa.

It is noted that an Electronic report is compulsory for the purposes of fulfilling the requirements provided by the client (by the project). Therefore both the Summary and Executive report will be electronically available and generated by using a special feature that the system will provide. (Look at FIGURE X to view process of generating an electronic report).

Below are the relevant sections outlining the technical details as well as visual representations of reports that will be generated.

6.1. OCCUPANCY REPORT

The occupancy report will be the report that will implement data from the database to produce usable analysed data that can be used for business decisions in the future. For instance, finding out the dates during which guests are arriving at the hotel more often to arrange for additional services such as decorations, games and activities during that time to entertain the guests.

6.1.1. DETAILED OUTPUT REQUIREMENTS

Below are the detailed output requirements.

TYPE	DESCRIPTION
OUTPUT TYPE AND ID	Occupancy report

REPORT OBJECTIVES	High-level overview of a subset of business activity
AUDIENCE	Middle-management (on hotel-basis), senior management
CONTENT	Occupancy levels for a period (monthly, quarterly, semi-annually, annually)
LAYOUT	Bar graphs showing the occupancy levels per month, pie charts showing the occupancy levels for each hotel for a single period
SELECTION	Click events
SEQUENCE	Sorted by date/time
COMPARISON	Occupancy levels from current period vs. previous periods
GROUPING	Concise paragraphs accompany each diagram
MEDIA	Electronic
FREQUENCY, TIMING, DELIVERY	On request
DISTRIBUTION	Through intranet or email
PRIVACY, SECURITY, INTEGRITY REQUIREMENTS	Restricted to internal access, high-confidentiality, reliable data.

#langanani RestEasy Occupancy Levels: 1st Quarter, 2017





Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

FIGURE 14: Occupancy report of a single RestEasy branch - for use by senior and middle management in strategic decision-making on a per-branch basis.


Logo

date and other info


Occupancy Report: startRange to endRange

Period Totals

Branch	Booked	Unbooked	Total	Average Occupancy	Average Rate	Average per Avail	Average Per Day	Revenue

Month 1 Figures

Branch	Booked	Unbooked	Total	Average Occupancy	Average Rate	Average per Avail	Average Per Day	Revenue

Month n Figures

Branch	Booked	Unbooked	Total	Average Occupancy	Average Rate	Average per Avail	Average Per Day	Revenue

Overview and comments

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

FIGURE 15: Occupancy report of a single RestEasy branch drilled out month and periods - for use by senior and middle management in strategic decision-making on a per-branch basis.

Occupancy Report - 1/12/2017 until 21/12/2017
 163083 2017/12/25 12:00:00 AM 2012/12/26 12:00:00 AM
 463362 2017/12/25 12:00:00 AM 2012/12/26 12:00:00 AM
 724105 2017/12/01 12:00:00 AM 2012/12/11 12:00:00 AM
 Days in the period:20

FIGURE 16: Occupancy report electronically generated using database information that can be passed into tableau by converting into CSV file.⁸

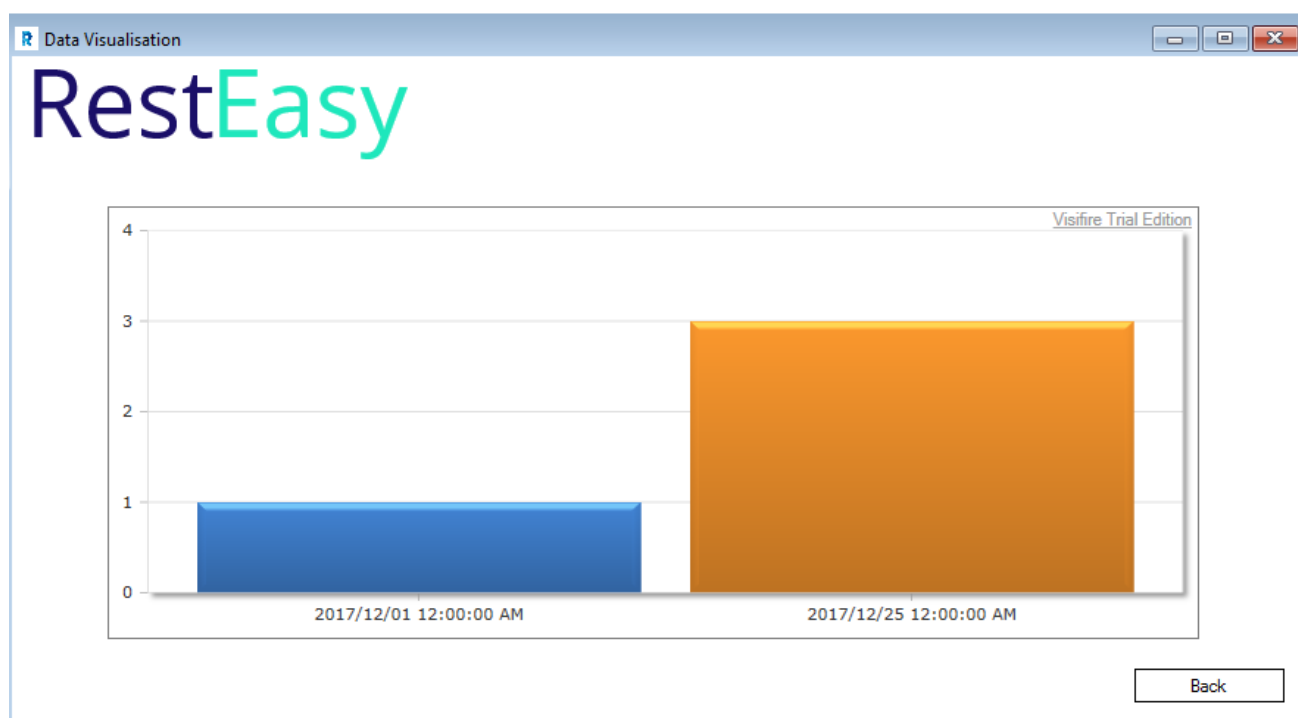


FIGURE 17: Data visualisation of occupancy levels based on the different dates.

6.2 DETAILED REPORT

6.2.1. DETAILED OUTPUT REQUIREMENTS

TYPE	DESCRIPTION
OUTPUT TYPE AND ID	Detailed report
REPORT OBJECTIVES	Provide exhaustive set of data for in depth machine learning powered analysis.
AUDIENCE	External Artificial intelligence APIs and Data Scientists (Analysts)

⁸ Tableau is a data analytics software application that can do in depth analysis of the data that is passed in CSV form, which is what can be generated by the database.

CONTENT	Individual transactions and database records
LAYOUT	Each record on a new line
SELECTION	By requestor by date
SEQUENCE	Sequential order
COMPARISON	Current group performance vs performance from previous period
GROUPING	No grouping, raw data
MEDIA	PDF
FREQUENCY, TIMING, DELIVERY	On request, delivered immediately
DISTRIBUTION	Compiled by lower-management and finance department
PRIVACY, SECURITY, INTEGRITY REQUIREMENTS	Access restricted to management. Confidential. Information verified and assessed before being deployed.

6.2.2. REPORT LAYOUT

76456 1 Deposit 99.50
 431476 1 Deposit 99.50
 508757 1 Deposit 99.50
 561405 1 Deposit 99.50
 565380 1 Deposit 99.50
 571112 1 Deposit 99.50
 856200 1 Room Account 2000.00
 Total payments:7

FIGURE 18: Report layout for detailed report

7. INPUT-OUTPUT STANDARDS & CONTROLS

The main focus of this section is to exhaustively identify the most necessary formalized inputs and outputs relative to the user. The focus will also be on the security to restrict access of critical data items, as well as verification processes for additions, deletions and updates of critical data. Data integrity has to be maintained and to ensure that, the input has to be validated and controlled well.

7.1. FORMALISED OUTPUTS

Below are the formalized forms of output included in the system.

1.1	FORMALISED OUTPUTS
1.1.1	Pop-ups and dialogs indicate successful and unsuccessful operations
1.1.2	Important user actions are coupled with feedback by the system that inform the user what they have done

7.2. BUILT-IN VALIDATION TO ENSURE REQUIREMENTS ARE MET

Below are the validation controls built into the system to optimize the performance of the application and to minimize the possibility of user error to ensure high performance and an enjoyable experience.

2.1	VALIDATION CONTROLS
2.1.1	Where possible, user data is automatically generated to prevent errors
2.1.2	Cache values into the system for common tasks
2.2	INPUT INTEGRITY CONTROLS
2.2.1	Input (symbols and combination of characters) into the system is verified before submitting
2.2.2	FIELD COMBINATION CONTROLS
2.2.2.1	Valid date ranges will be checked - a user will not be able to enter a ending date that is before the start date
2.2.2.2	Cannot enter a date that has already been passed (Cannot book for a date that has already passed)
2.2.3	VALUE LIMIT CONTROLS
2.2.3.1	Cannot book for an extended period time without proper approval from higher management.

2.2.3.2	Number of guests need to be of a certain limit (Controlled with combo boxes and spinners) unless approved by higher management
2.2.4	COMPLETENESS CONTROLS
2.2.4.1	The system will inform the user whether fields are optional or necessary as well as informing them whether the data entered is correct or not (re-entering passwords)
2.2.4.2	Can use a calendar control to help with completing the typing of a date and assisting user with dates and times of booking
2.2.5	DATA VALIDATION CONTROLS
2.2.5.1	Where possible, input will be automatically filled to prevent user error
2.2.5.2	Using escape characters to prevent SQL injection from hackers, for example to prevent ticks from being inserted into dynamic requests into the system
2.2.5.3	The system will use different types of components such as spinners and combo boxes to prevent invalid data from being entered into the system
1.2.5.4	Outliers need to be verified before being submitted into the system

3	OUTPUT INTEGRITY CONTROLS
3.1	DESTINATION CONTROLS
3.1.1.1	Requested information will be sent only to the people requesting the data and those that they specify, permitting that the requester and those specified have authorization to access the information
3.1.1.2	Encrypt confidential information so that it cannot be accessed
3.2	COMPLETENESS, ACCURACY, CORRECTNESS CONTROL
3.2.1	Information will be time-stamped automatically with the date of issue and the creator
3.2.2	The page number will be displayed on each page of every document generated
3.2.3	Remove outliers such that information that is inaccurate and outside bounds is flagged and hence notified to the managers so that action can be taken to improve accuracy
3.2.4	Beginning header with report identification and description can be automatically generated - filename, headings, user and current date

8. IMPLEMENTATION PLAN

Below is the detailed implementation plan scheduling the tasks that have to be completed over the design and the implementation phases.

MILESTONE 1

Planning, Specifications, Design

This milestone achieving path will entail the start of the project, formatting of documents, analysis of the full requirements, sign-off from technical including algorithms and data structure. All deliverables will be to the satisfaction of RestEasy.

The first deliverable for this milestone is to ensure that all involved executives in the process will sign off their documents as well as the legal agreements.

DELIVERABLE: SIGNED OFF DOCUMENTS AND LEGAL AGREEMENTS.

	PROJECT IMPLEMENTATION PLAN
1.1	TECHNICAL DOCUMENTATION
1.1.1	User requirement document (Including business requirements and product and technical requirements)
1.1.2	A log of user stories to explain how the end user is intended to use the system in general and each of its features in particular.
1.1.3	Access to live source code - the source code may be stored in collaboration repositories such as github.com or bitbucket.org.
1.1.4	Running code (.exe file) - the working version of the program that is also stored on collaboration platforms such as github.com
1.1.5	"ReadMe" file-detailed description of the system installation process and troubleshooting
1.1.6	Host server login credentials.
1.1.7	Bugs and issues documentation-compilation of all found bug reports created at a QA and testing stage, as well as new issues reports at a post-release stage.
1.2	DESIGN DOCUMENTATION

1.2.1	UI design & wire-frames, mockup's / UI Designs that visually outline the system's functionality and technical needs.
1.2.2	Style guide guidelines for usage of colours, fonts, and other graphical elements of app design.
1.2.3	Splash screen and other welcome screens (system icons, logos and other designed elements.)
1.3	LEGAL DOCUMENTATION
1.3.1	Memorandum of Understanding with interested parties
1.3.2	Term sheet agreements with interested parties

EXPECTED COMPLETION AND BUDGET

Date : 30-Oct-17

Amount needed : R212, 619.00

MILESTONE 2

DEVELOPMENT, TESTING

Achieving this milestone will involve development with 3rd party software house.

The main deliverable to achieve this milestone would be to ensure that there is a working application ready for beta testing.

DELIVERABLE: WORKING APPLICATION READY FOR BETA TESTING.

START	PROJECT IMPLEMENTATION PLAN
2.1	System functionality development
2.2	Beta version or 1st build release
2.3	Testing - user acceptance testing and software assurance test results.
2.4	Submitting and deployment
2.5	bugs and issues documentation - compilation of all found bug reports created at a QA and testing stage, as well as new issues reports at a post-release stage.
2.5	Update on technical and design documents

Achieving the milestones have been laid out however there will be some drawdown conditions that will actually affect the achievement and completion of the milestones in a successful manner. Below are the drawdown conditions.

DRAWDOWN CONDITIONS:

- Signed audit report for the successful completion of Milestone 1 which must be to the satisfaction of RestEasy.
- Signed request from the recipient for the payment amount of Milestone 2 including the breakdown of project costs and confirmation of recipient banking details.
- Proof of contribution by the recipient to complete Milestone 2.

EXPECTED COMPLETION AND BUDGET

At the end of the day, all deliverables have to be to the satisfaction of RestEasy.

Date : 02-Jan-18

Amount needed : R 106,309.50

MILESTONE 3**CERTIFICATION, DEMONSTRATION, DOCUMENTATION**

This will entail testing, and debugging processes and piloting.

DELIVERABLE: WORKING APPLICATION READY FOR ROLL OUT AND ALL AGREEMENTS IN PLACE.

START	PROJECT IMPLEMENTATION PLAN
3.1	Testing - user acceptance testing and software assurance test results.
3.2	Submitting and publishing on application store
3.3	Updated on technical and design documents

DRAWDOWN CONDITIONS:

- Signed audit report for the successful completion of Milestone 2 which must be to the satisfaction of RestEasy Management.
- Signed request from the recipient for the payment amount of 100% of Milestone 3 including the breakdown of project costs and confirmation of recipient banking details.
- Proof of contribution by the recipient to complete Milestone 3.

EXPECTED COMPLETION AND BUDGET

All deliverables to be to the satisfaction of RestEasy.

Date	: 02-Jul-18
Amount needed	: R 106,309.50
Total project cost	: R 531,547.50

9. TEST PLAN

Testing is extremely vital to building a fully functional and efficient system. This section will cover the test environment, test items, test approaches, test cases and the test schedule.

9.1. TEST ENVIRONMENT

Below are the explicitly, and straight to the point hardware and software requirements. After this is the low level specifics of the hardware and software requirements.

- All hotels should implement hardware and software consistently and ensure good integration.
- All hotels should implement Windows 10 (Professional edition) as the main operating system.
- Coding language used to develop system is C#.net.
- C# is compiled into MSIL (Microsoft Intermediate Language) for .net apps.
- Microsoft Visual studio will be used as the integrated development environment to build the application.
- SQL will be used as the main Database Management system integrated into Visual Studio.
- SQL express can be used for generating the ER diagram. However, due to various complexities, a drawing tool can always be used in collaboration with SQL express to create the ER Diagrams.

The system will be tested against its minimum and recommended hardware and software specifications which are as follows:

MINIMUM SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE	Single User
Minimum	Pentium III
Recommended	Pentium IV 2GHz or Higher
OPERATING SYSTEM	
Minimum	Windows XP
Recommended	Windows XP or higher
RAM	
Minimum	512 MB
Recommended	1.028 GB or Higher
SCREEN RESOLUTION	

Minimum	800 x 600
Recommended	1024 x 768

9.2. TEST ITEMS

Below are the items that will need to be tested rigorously.

1	BOOKING SUBSYSTEM
1.1	Stress testing where the system is in high use (i.e. peak seasons)
1.2	Correct number of rooms reserved from number of guests input
1.3	Correct room allocation
1.4	Unavailable date is unelectable
1.5	Missing input entries
1.6	Duplicate information - ID/passport, email
1.7	Booking start date precedes booking end date
2	E-COMMERCE TRANSACTING
2.1	Correct reflection under guest accounts
2.2	Appropriation of funds
2.3	Completion of transactions
3	REPORTING SUBSYSTEM
3.1	Correct data and information is retrieved from system database. Includes checking data integrity controls and validation techniques.
3.2	Correct information
3.3	Correct report type
3.4	Valid period entered
4	ACCOUNTING SUBSYSTEM
4.1	Legal regulations met in system operation

4.2	Transactions reflect in accounts
4.3	Entries added to correct accounts
4.4	Transaction not reflected
4.5	Correct taxation on relevant transactions

9.3. TEST APPROACHES

FIRST TESTING IMPLEMENTATION TO BE DONE BEFORE DEVELOPMENT AND REVISED AFTER BETA BUILD COMPLETED

1. USER ACCEPTANCE TESTING
Involves versions of the system being created and handing them to users that it is intended to be used by. These users assess and verify that the system meets their requirements.
2. SOFTWARE ASSURANCE TESTING
Monitors the progress of the development process and reiterates system fundamentals to be used in development such as methodologies and good project management. Developers are encouraged to follow recommended principles and processes so that the strategic goal of develop is attained and weaknesses in the development process can be acted upon so that system deliverables are error-free, consistent and maintainable.
3. UNIT TESTING
Creating test cases for methods to ensure that they are properly functionality.

9.4. PROBLEM TRACKING (TEST CASES)

Describe the process followed for tracking and resolving errors

1	BOOKING SUBSYSTEM
1.1	<i>Case:</i> guest makes reservation <i>Data:</i> stay period, number of guests, suite type, check-in time <i>Returns:</i> populated reservation object
1.2	<i>Case:</i> create guest account <i>Data:</i> guest name and surname, ID/passport number, email address, age <i>Returns:</i> populated Guest object and entry into database

1.3	<i>Case:</i> select stay period <i>Data:</i> stay start date, stay end date <i>Returns:</i> void - initialises stayStart and stayEnd attributes
2	E-COMMERCE TRANSACTING
2.1	<i>Case:</i> transfer funds <i>Data:</i> credit card number, CVV number, PIN <i>Returns:</i> transaction status
3	REPORTING SUBSYSTEM
3.1	<i>Case:</i> create report <i>Data:</i> start period, report period (<i>month, quarter, semi-annual, annual</i>), report type <i>Returns:</i> Report object
4	ACCOUNTING SUBSYSTEM
4.1	<i>Case:</i> capture transaction <i>Data:</i> amount, record type, GuestID <i>Returns:</i> void, entry committed to database

9.5. TEST SCHEDULE

Description of how the test schedule was co-ordinated with the development schedule

NO.	TEST SCHEDULE	PROJECT STATUS
1.1.1	User acceptance testing and software assurance test results.	Beta release - first deliverable
1.2.1	User acceptance testing and software assurance test results.	Build 1.0 - final deliverable