Section 2 - Investigation

Table of contents:

Discussion of the nature of the solution

The general problem my system is trying to solve is that hotels need an effective way to allow customers to book rooms at their hotel and an easy way to display this data. As my system is a hotel booking system this makes it very commercially viable as it can be used by any hotel. My system could even be used by other organisations in the hospitality industry for example B&Bs as well as caravan parks (each caravan could be considered as a room). Although my target user will be hotels as these are the largest organisations in the hospitality industry. The hospitality industry is very large and therefore there are many customers that are able to purchase my system.

To identify similar products as mine I will simply search the term ‘hotel booking system’ on a search engine and select certain websites. When I have selected a website I will look at the website to see if they are advertising themselves as hotel booking software. I will be using the two main search engines (Google and Bing) as these will provide a wide range, and accurate, set of results as they are so large.

I will identify similar existing systems by downloading free trials of the software and checking if their core functionality is the same as mine. I will also use search engines like Google or Bing to search for hotel booking systems and see what these systems offer. By using these search engines I will also be able to find websites that review this type of software which will allow me to find customer reviews and find out what the systems are missing.

Research:

Cloudbeds

Inputs, processes and outputs

- One process which Cloubeds offer is that it will update reservations made on the website to the database automatically.
- The reservations are displayed as a tape chart instead of in a tabular form.

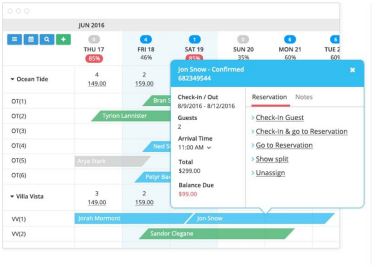
Functionality

- One piece of functionality which cloudbeds offers is that it can link with other websites, such as booking.com and expedia to allow customers to book a room on these websites. I will not be able to add this functionality into my system but I will be creating a user interface the customer can use to book a room.
- Another piece of functionality is that it is possible that it is possible to check what customers have paid or not. This will be incorporated within my system.

Limitations

- It is a web based application and therefore requires an internet connection to use.
- Another limitation of cloudbeds is that it is very difficult to edit reservations that has already passed.
- A complaint about cloudbeds is that there is a lot of ‘clicking’ as most features require the user to start at the calendar page.

Screenshots:



eZee Frontdesk

Inputs, processes and outputs

- When logging on to the system a drop down list of the usernames is used instead of the user manually typing out their username. Depending on the amount of users this may or may not save time as if there are few users it would be easier. But if there are many users it would take them more time to look through the list.
- The colour of the interface and names are customizable.

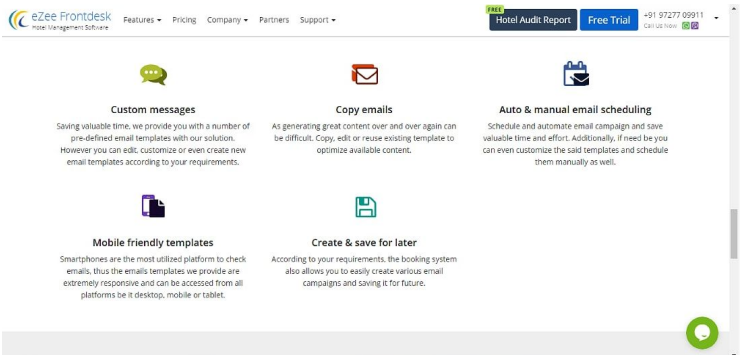
Functionality

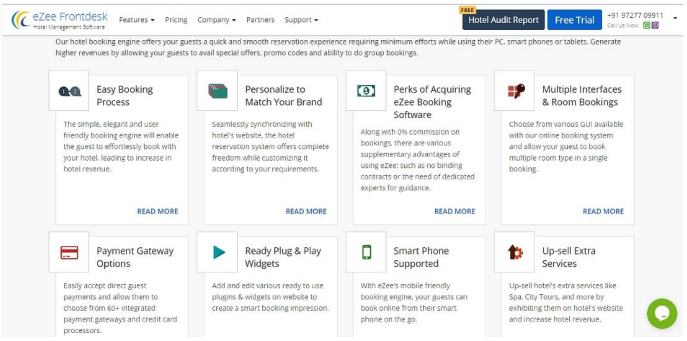
- Automatic emails can be set up and it also provides templates for these automatic emails.

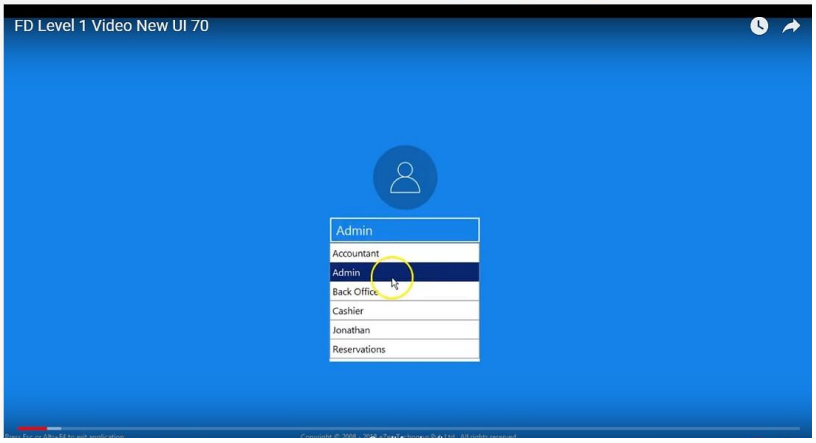
Limitations

- The reservation and check-in screens are seperate.
- For online reservations the user has to click on the reservation to check if it has been assigned a room.

Screenshots







Frontdesk Anywhere

Input, process and output

- The output of the reservations is of a calendar and not in a tabular form.
- All data is encrypted and automatically backed up.

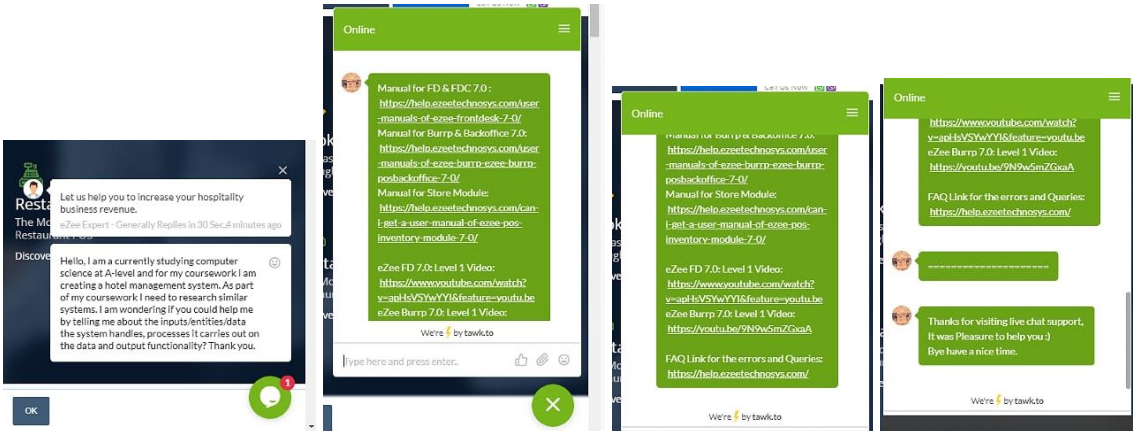
Functionality

- The main user is able to create an unlimited amount of other users and can create user levels with restrictive permissions.
- Another piece of functionality that this system has is that it is able to send customized emails to guests.

Limitations

- A problem with the functionality of this system is that automatic thank you letters cannot be sent with the hotel logo.
- Another problem with this system is that users cannot search for reservations by customers email or phone number.
- Programming experience is required to send customized emails.
- Cannot send bulk customized emails.

Screenshots



Hilton

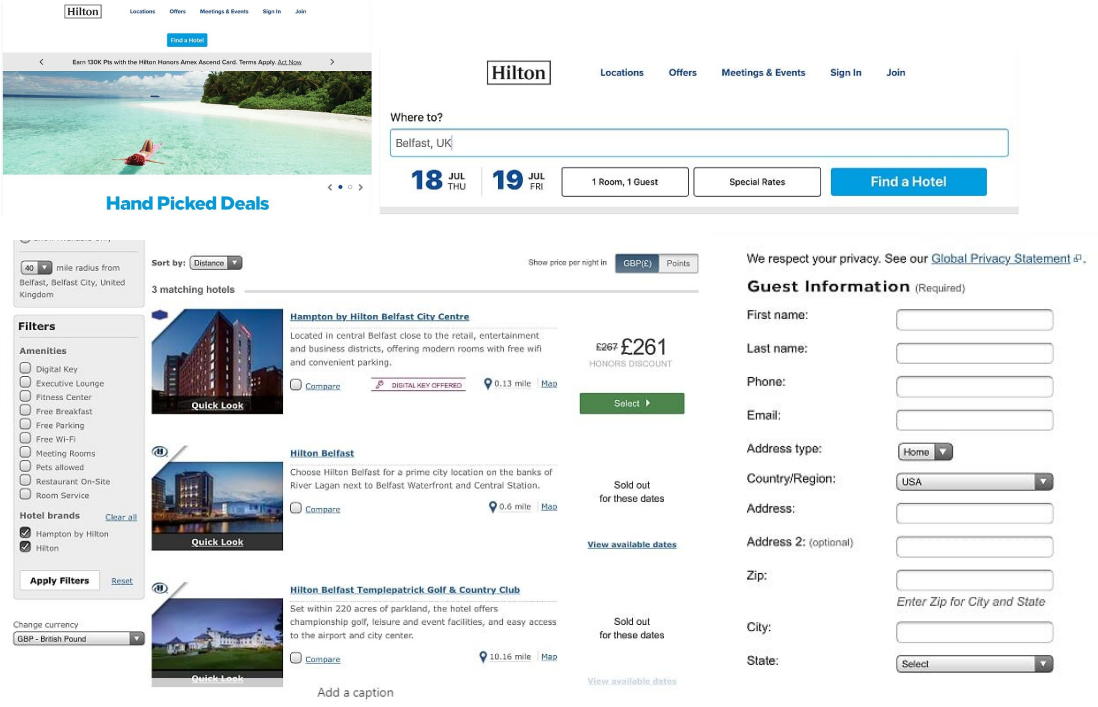
Input, process and output

- The customer first has to input how many rooms they would like to book and how many people are there per room. The customer then has to specify how many people staying in the room are children and how many are adults.
- The information which Hilton asks for its customers is their first name, last name, phone number, address, email and country.
- The payment information that is stored is the card number and the expiration date.

Functionality

- There is a scroll down list of the rooms which allows users to select a room.
- There is an option to input any ‘special requests’ that a customer might have.

Screenshots



Sirvoy

Inputs, processes and outputs

- This system allows users to customise the input fields for the guest.
- This system also allows automatic emails to be sent out.
- Guests can also input discount codes to get money for their stay.

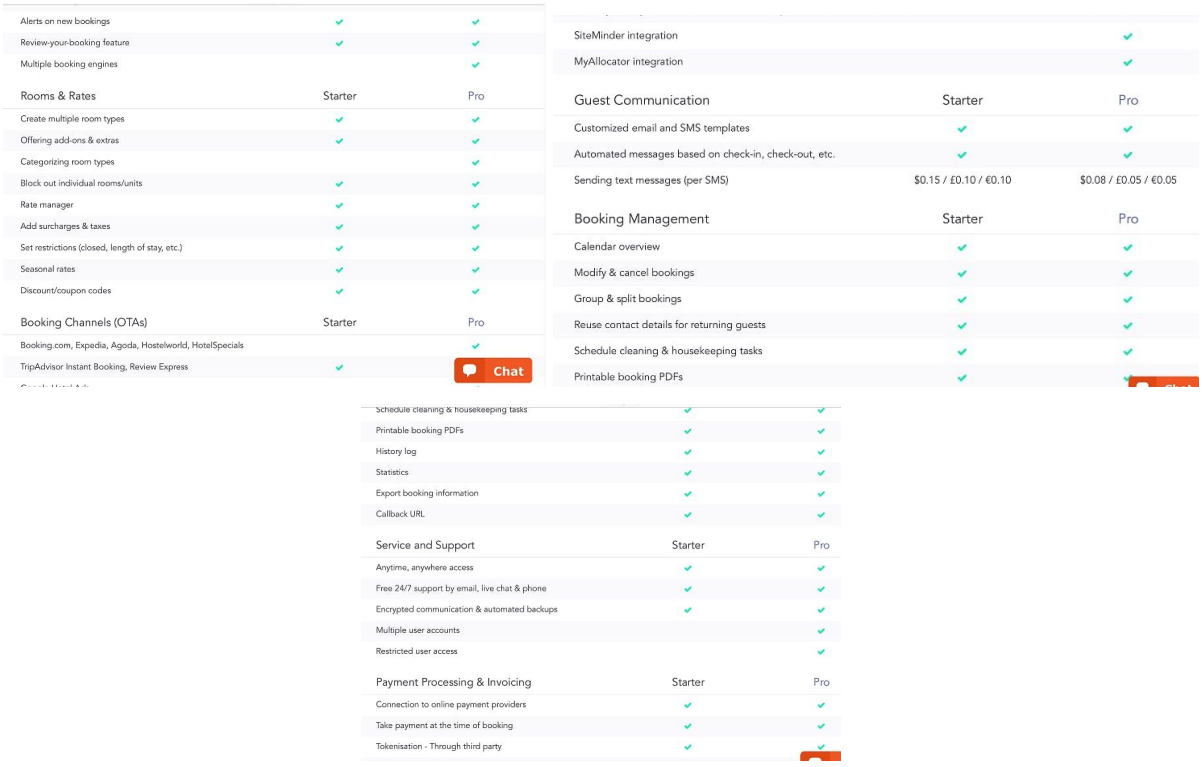
Functionality

- The main user of the system is able to create different room types.
- There can be multiple user accounts which can have different levels of restrictive access.
- The data on the system is also encrypted.

Limitations

- Complicated terminology used within the system.
- The system does not allow bookings to be over 1 and a half years in advance.
- The user cannot add attachments to a particular room.

Screenshots



Alerts on new bookings	✓	✓
Review-your-booking feature	✓	✓
Multiple booking engines		✓
Rooms & Rates	**Starter**	**Pro**
Create multiple room types	✓	✓
Offering add-ons & extras	✓	✓
Categorizing room types		✓
Block out individual rooms/units	✓	✓
Rate manager	✓	✓
Add surcharges & taxes	✓	✓
Set restrictions (closed, length of stay, etc.)	✓	✓
Seasonal rates	✓	✓
Discount/coupon codes	✓	✓
Booking Channels (OTAs)	**Starter**	**Pro**
Booking.com, Expedia, Agoda, Hostelworld, HotelSpecials		✓
TripAdvisor Instant Booking, Review Express	✓	

Chat

SiteMinder integration		✓
MyAllocator integration		✓
Guest Communication	**Starter**	**Pro**
Customized email and SMS templates	✓	✓
Automated messages based on check-in, check-out, etc.	✓	✓
Sending text messages (per SMS)	$0.15 / £0.10 / €0.10	$0.08 / £0.05 / €0.05
Booking Management	**Starter**	**Pro**
Calendar overview	✓	✓
Modify & cancel bookings	✓	✓
Group & split bookings	✓	✓
Reuse contact details for returning guests	✓	✓
Schedule cleaning & housekeeping tasks	✓	✓
Printable booking PDFs	✓	✓

Schedule cleaning & housekeeping tasks	✓	✓
Printable booking PDFs	✓	✓
History log	✓	✓
Statistics	✓	✓
Export booking information	✓	✓
Callback URL	✓	✓
Service and Support	**Starter**	**Pro**
Anytime, anywhere access	✓	✓
Free 24/7 support by email, live chat & phone	✓	✓
Encrypted communication & automated backups	✓	✓
Multiple user accounts		✓
Restricted user access		✓
Payment Processing & Invoicing	**Starter**	**Pro**
Connection to online payment providers	✓	✓
Take payment at the time of booking	✓	✓
Tokenisation - Through third party	✓	✓

skyTouch

Inputs, processes and outputs

- The reservations are stored as a tape chart.
- The maintenance staff has a different user interface than the reception staff.

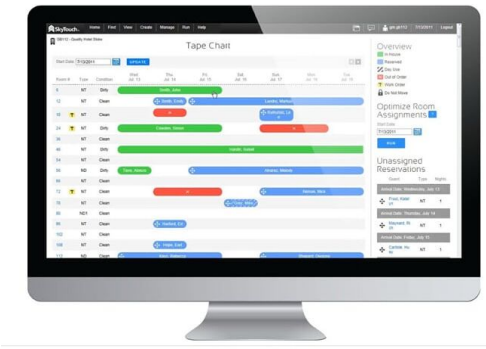
Functionality

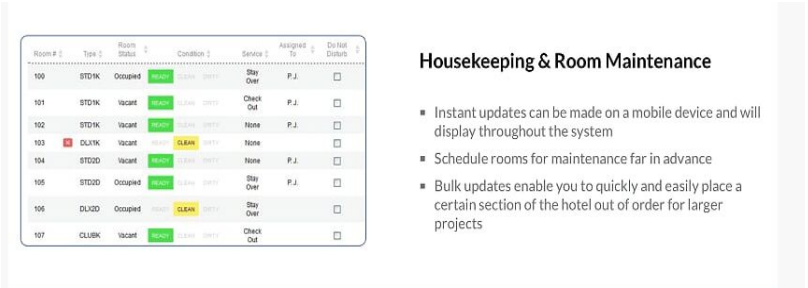
- This system will automatically schedule which rooms maintenance should go to at which times.
- The system will store how many nights a guest stays at the hotel and will offer the guest discounts and rewards depending on this.

Limitations

- Users cannot access past information i.e. customers that have already stayed in the hotel or any other information about last bookings.
- This system is web-based and therefore requires an internet connection.

Screenshots





MSI CloudPM

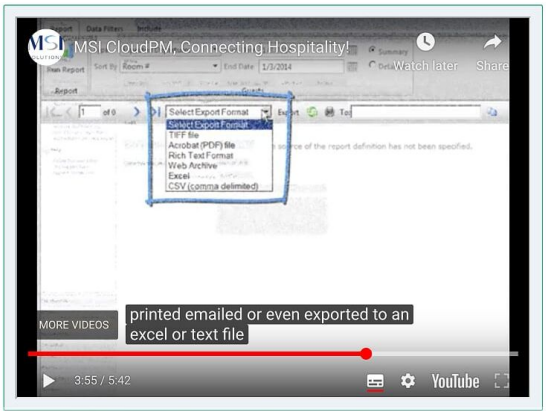
Inputs, processes and outputs

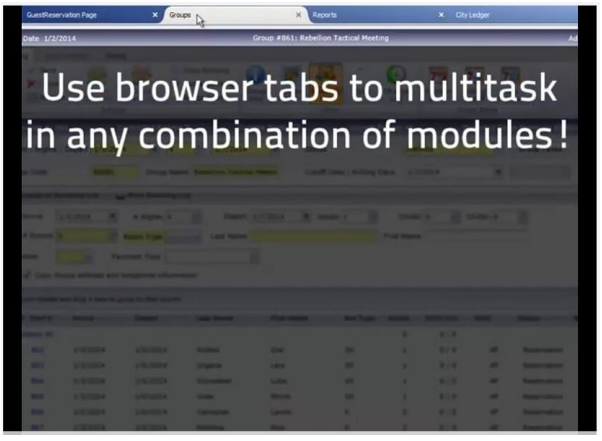
- The user interface of this system includes tabs that allows the user to switch to different windows in the system.
- In this system it is possible to print, email and export the database as an excel file.

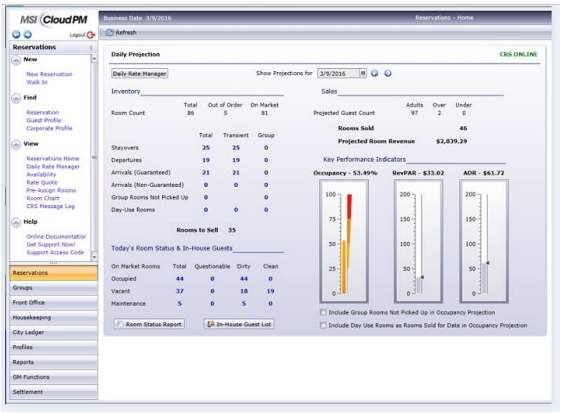
Functionality

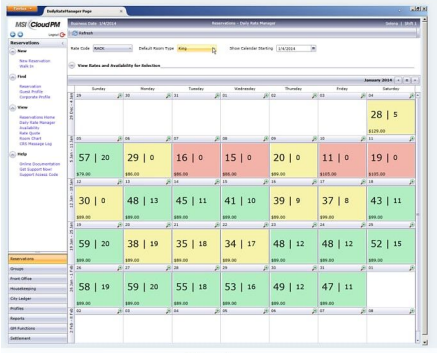
- This system encrypts the database using tokenization technology.

Screenshots









OPERA Property/Oracle

Inputs, processes and outputs

- This will record the guests preferences and allows reception or other staff members to view these.

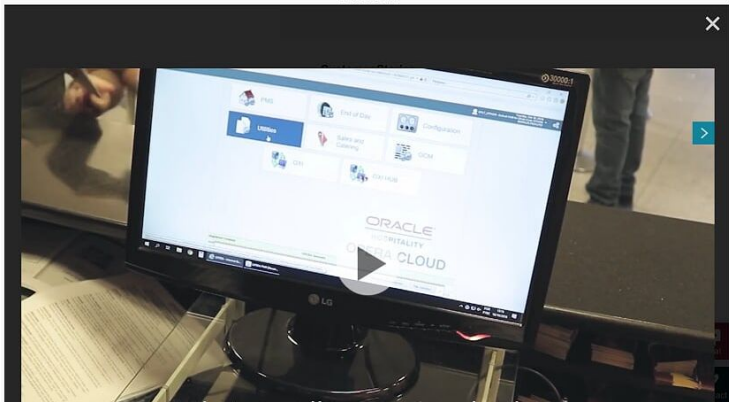
Functionality

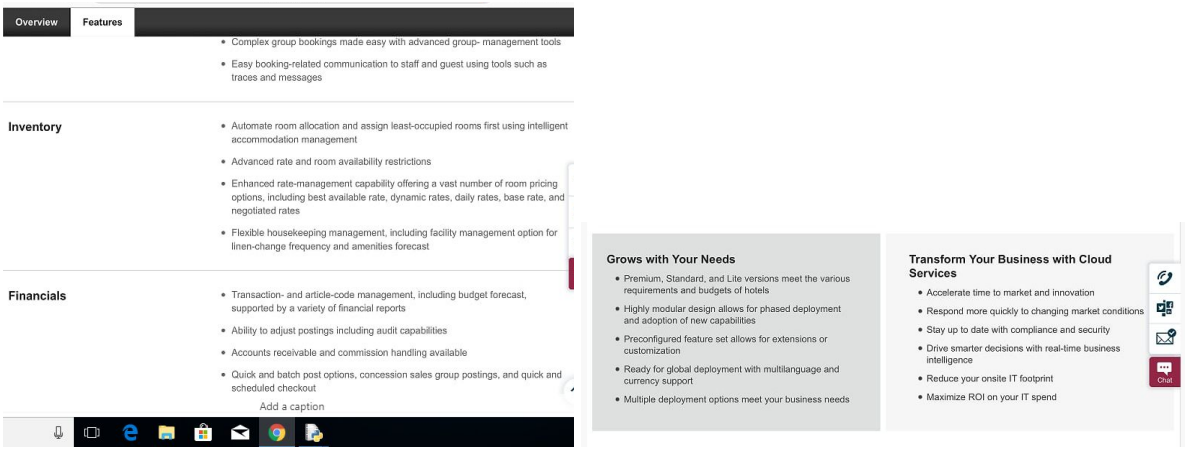
- Hotel staff is able to check guests in and out of their rooms.
- This system will assign the rooms that are least occupied before the more popular rooms.

Limitations

- It is not very customizable for the user.
- There are too many pop ups on the system.

Screenshots





Hotelogix PMS

Inputs, processes and outputs

- Customers are able to leave a review of the hotel on the system.

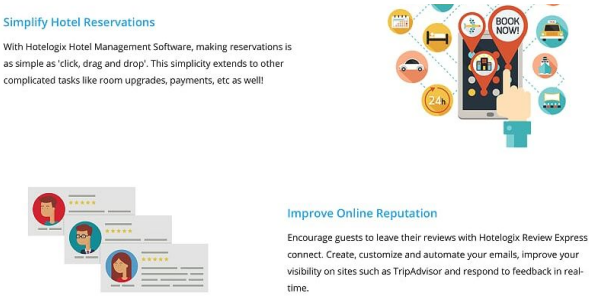
Functionality

- This system has built in accounting software.

Limitations

- Labels for certain features are confusing.
- It does not send reminder emails to guests a few days prior to the guests arrival.

Screenshots

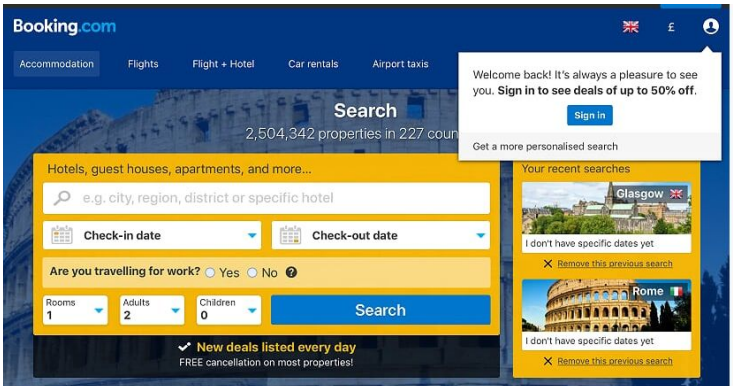


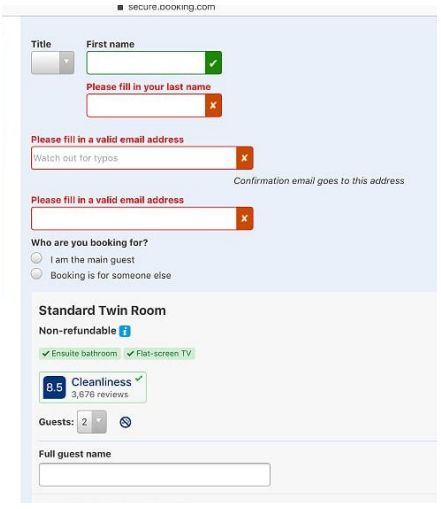
Booking.com

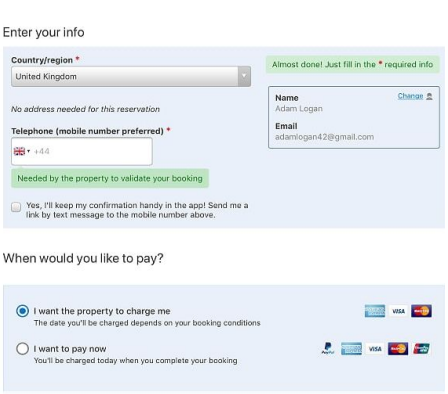
Inputs, processes and outputs

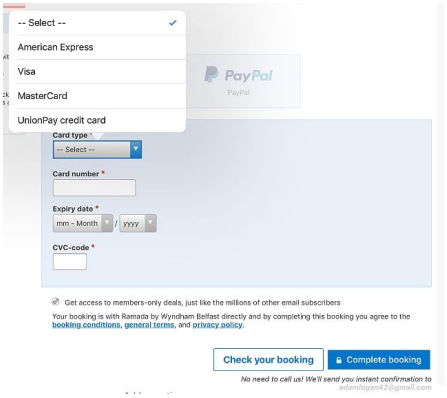
- There is an option available that allows the guest to select if they are travelling for work or not.
- The guest can also select the check and check out dates themselves.
- There is also an option for the user to select how many rooms they would like to book at this time.
- When selecting how many people there are the guest has to select if they are adults or children.
- The system also asks the user if they are the main guest or if they are booking on behalf of someone else.
- The user is also prompted if they have any special requests and what country they are from.
- The user is also asked for their mobile number.
- The card details that are requested is the card type, the card number, expiry date and the CVC-code.

Screenshots









Final list of functionality

From the research above, I have decided that I will include the functionality below:

- The data that is stored on the database will be encrypted. This functionality can be found in ‘MSI CloudPM’, ‘Frontdesk Anywhere’ and ‘Sirvoy’.
- The database will be backed up. This functionality is found in ‘Frontdesk Anywhere’.
- There will be an option that will allow the system to send automatic confirmation and thank you emails to guests. An email will also be automatically sent to the guest a few days before their arrival to remind them of the booking. This functionality can be found in ‘eZee Frontdesk’, ‘Sirvoy’ and ‘MSI CloudPM’
- It is possible to set user access levels to restrict certain users. This piece of functionality can be found in ‘Frontdesk Anywhere’.
- The main user of the system will be able to create different room types and will be able to link these room types to each room. This is functionality that has not been specified by any system through my research. I have decided to include this as rooms may be renovated and therefore the database needs to be updated.
- The ability to store the amount of nights a guest stays. The first part of this statement was found in all the systems I have investigated.

Summary specification

The overall goal of the project is to create a system that will assist hotel owners in managing their hotels. This includes making it easier for customers to book rooms for the hotel and to show the manager what rooms in the hotel are least popular. My system will also tell maintenance staff and cleaners what times the rooms are free so they can either clean or fix the rooms. The purpose of the system for the hotel owner is to make their hotel more efficient and to reduce costs in their hotel.

Broad aims:

- To reduce costs in the hotel.
- To produce a hard copy of the bill for the customer.
- To give the maintenance and cleaning staff the times that rooms are vacant.
- To increase the number of bookings in the hotel.
- To allow the hotel owners to check what is going on in their hotels.

Requirements for the proposed system

- The customer/guest:
    - Functional requirements:
        i. The first functional requirement the customer has is to be able to book their room on the system.
        ii. The system should also be able to accept the customers credit and debit card information.
        iii. Another functional requirement is that the system should display the rooms that are available to the customer at their specified time.
        iv. The customer will be able to leave a comment after they stay for improvements on the hotel.
        v. The customer will also receive an automatic confirmation email after they book their stay.
    - Non-functional requirements:
        i. The credit and debit card information provided by the customer should be validated and it should also be encrypted as it is very sensitive information.
        ii. The customer will be able to view, edit and add their record on the customer entity. They will also be able to view their record in the booking entity, room entity, room type entity, the holiday type entity and the bill entity. They will not be able to access any other records except form their own. This reason for this is for the privacy of the other customers as they may wish for their information not to be known by people who are not hotel staff.
- The receptionist:
    - Functional requirements:
        i. The receptionist should be able to book rooms for customers if asked.
        ii. Another functional requirement is for the receptionist to be able to print out the customers bill if requested.
        iii. The system should be able to tell whoever it concerns which rooms are occupied and who occupies these rooms.
        iv. Receptionist staff should also be able to add customers to the customer table.
        v. A functional requirement of the system is that it should allow a receptionist to edit the bill table in case there are any mistakes which the customer complains to the receptionist about.
    - Non-functional requirements:
        i. One requirement is that only receptionist will be able to access this information and therefore they will need a strong and secure password.
        ii. The system should have enough storage so it can be able to store information on every room in the hotel.

iii. As the receptionist can view information on every room a scrollbar will need to be implemented so they can scroll through all the rooms in the hotel.

- The management staff:
  - Functional requirements:
    i. They should be able to view and edit the staff table as they would be in charge of change in staff and pay rises which they can edit on the system.
    ii. Management staff should be able to edit, delete and add the login details of all users on the system, the customer records and each booking.
    iii. Another functional requirement is that management staff can view all tables in the database.
    iv. Management should also be able to edit the automatic emails that are being sent out to guests.
  - Non-functional requirements:
    i. The management staff should have the most secure accounts as they have access to the entire system. They will need a more complex encryption algorithm than the other users to protect their passwords. Another possible solution is to only allow the management staff to have secure passwords. A secure password is one with symbols, numeric characters, upper and lower case characters.
    ii. The system should be able to provide an output of a graph which shows the bookings by month and a hard copy of the customer report.
- The cleaning staff:
  - Functional requirements:
    i. The cleaning staff will need to be able to view the times when guests check-in and check-out so they know when to go and clean those rooms.
  - Non-functional requirements:
    i. The system should automatically send a notification to the cleaner when a room needs to be cleaned whether the room is occupied or not.
- Maintenance staff:
  - Functional requirements:
    i. The maintenance staff should be able to check which rooms are occupied so they know when they have to do their maintenance work.
    ii. Another functional requirement is that the maintenance staff should be able to mark jobs as complete when they have finished the job.
  - Non-functional requirements:
    i. An email should be automatically be sent to the maintenance whenever something has been broken along with the date and time when the maintenance staff will be able to fix the problem.
- Accountancy staff:
  - Functional requirements:

i. The accountancy staff should be able to view the bookings, bills and staff tables. This will give them access to all the financial data in the database.
ii. Another requirement of the accountancy staff will give the total amount of money spent on labour.
iii. The system will have a built in calculator that will perform simple calculations i.e. addition,subtraction, multiplication and division.

- Non-functional requirements:
  i. One non-functional requirement of the accountancy staff is that their passwords should be very secure or otherwise hackers could gain access to very sensitive data.

Methods to be used in the system

When creating my system I will be using many programming tools and one of the main programming tools that I will be using is an if statement. The reason for this is because with an if statement the program can make decisions about what code that needs to be executed depending on the condition. There will be many occasions where I will need to do this and an example of one of these occasions is when I need to compare two items in the database.

Another programming tool which I will be using is loops. The reason why I will be using loops is because it will be more efficient rather than typing out the same code several times. Loops will also make my code easier for other developers to understand. There are many different types of loops but the two main loops that I will be using is the for loop and while loop. The reason why I will be using a for loop is because these can iterate through arrays/lists very easily. The other loop which I will primarily be using is the while loop as this will help with validation as it can continue to prompt the user for an input until it is valid.

A programming tool which I will be using exclusively for validation is the try except block. As the name suggests it will attempt to execute a piece of code but if a specified error occurs then it will execute the piece of code within the except section is the code. I will also be using this programming tool to troubleshoot my code so I can find errors easier, where they are within the code and what exact error is causing the code to crash or not work as intended.

I will also be using user defined subroutines. This programming tool will allow me to execute a section of code several times without having to type it out every time I use it. As with loops this makes my code easier to understand as code is not being repeated. This will also make my code more efficient as I am re-using the same section of code over again instead of creating different pieces of code that carry out the same task. This also allows me to troubleshoot my code easier as if an error occurs I only have to fix and troubleshoot the subroutine but if I did not use the subroutine then I would have to fix each point where I used the piece of code in question.

I will be leaving comments in code so other developers are able to understand my code. These comments will also assist me as if u leave the code for an extended length of time I

may forget what I have done. Comments will allow me to be able to remind myself of what I have done without having to go through each line of the code and work it out.

One library in Python that I will be using is the tkinter library. This library in Python will allow me to create a GUI (Graphical User Interface). This is crucial for my solution as I am planning to implement a GUI to make my solution more intuitive for inexperienced users. Within tkinter I will also be using another library called tview. This allows me to display the data in the database within a tabular format. Another function that can be used with tview is tabs. This function will create several tabs on the top of the window and allows users to switch through each tab. I will use this function by creating a tab for each entity which simply allows me to hide any tabs if a specific user cannot view the entity within that tab. An example of this is the guest will not be able to access the staff entity as this tab will be hidden. Within tkinter there are many functions including ones that allow you to create entry widgets. These entry widgets have many features including a feature that allows the text that is shown in the entry widget to be changes. This allows me to create a password entry widget that will display the text in the entry widget as '*' or any other symbol or letter which I choose.

Another library which python provides is called docs. This is an external library in Python so it will need to be installed through the command prompt using the command 'pip install docs'. This library allows docs files to be created and edited within python. This allows for the creation of a bill for a customer or if possible an option to print out the entire database. The files that are created can be saved in a specific directory so the files can be easily found. It could even be possible to allow the owner of the hotel to choose this directory. The hotel logo could also be inserted into these word documents so guests could easily identify that the bill is from the hotel.

I will also need to use the sqlite3 library to allow the manipulation of the database. Using this I will be able to use SQL code to edit, create and delete records. Another bit of functionality that this library contains is the ability to create entire entities and also create the relationships these entities have with each other. Using this library allows me to achieve the core functionality of my system which is to handle and create a database.

Two other libraries in python which I will need to use is the smtplib library and the email library. These libraries used together will allow me to send emails with attachments and links to websites. The smtplib library connects the user to a server which will send the email. A downside of using this library is that a specific server needs specified. This results in my program being limited in what emails it can accept. For example it may only accept gmail and outlook emails but it may not accept yahoo emails because I need to specifically request access to their servers. To create a subject the email library is used and this library is also used to create the body of the email using html. This library is also used for adding attachments to the email. These libraries can be used to send automatic thank you emails along with the bill as an attachment. The hotel should also be able to include an image of their hotel’s logo as html is used to create the body of the email.

The shutil library in python can be used to create a copy of the file. The purpose of this is so the database can be backed-up. The other libraries to backup a system are pickle and os. The pickle library allows the storage of the previous month when the file was backed up and therefore allows me to create a periodic backup. The OS will allow me to delete a file when there are too many backups.

Objectives/Success Criteria

1. Functional requirements:

1.1. Customer:

1.1.1. The system must allow the creation of records:

1.1.1.1. The system should allow the following stakeholders to create records:

1.1.1.1.1. Customer/guest. They should only be allowed to create one record.

1.1.1.1.2. Management staff.

1.1.1.1.3. Receptionists.

1.1.1.2. Validation:

1.1.1.2.1. The ‘CustomerID’ should be checked to see if it already exists.

1.1.1.2.2. The ‘Telephone_number’ field should have a length check as phone numbers can only be 12 digits long.

1.1.1.2.3. The fields ‘Frst_name’ and ‘Srn_name’ should both be checked if they contain non-alphabetical characters as names only contain letters.

1.1.1.2.4. The ‘BookingID’ and ‘BillID’ foreign keys will need to be checked if they exist in their respective tables.

1.1.1.2.5. ‘Allergies’ will need to have a lookup check as there is only a certain amount of allergies.

1.1.1.2.6. The ‘Age’ field will need to have a type check form of validation to check if it is an integer.

1.1.1.2.7. ‘Postcode’, ‘Email’, ’Expiry date’ and ‘Address’ should all have a format check as they all have to be within a certain format.

1.1.1.2.7.1. ‘Expiry_date’ will also be checked if it is in the future.

1.1.1.2.8. The ‘Payment_type’ field will be validated using a lookup check as there are only a certain amount of payment types the hotel can accept.

1.1.1.2.9. The fields ‘Card_Number’ and ‘CVC-code’ will be checked if they are integers.

1.1.1.2.10. ‘Username’ will be checked to see if it matches any other usernames in the system.

1.1.1.2.11. The ‘Password’ field will be checked if it is above a certain amount of characters, if it contains uppercase and lowercase characters and if it contains alphanumeric characters.

1.1.2. The system must allow the following stakeholders to read the customer table:

1.1.2.1. The customer/guest must only be allowed to read their record.

1.1.2.1.1. This will be outputted as a simple list on the system.

1.1.2.1.2. A confirmation email will be sent to the user.

1.1.2.2. The receptionist will be able to read the whole table.

1.1.2.2.1. This will be outputted as a table on the system.

1.1.2.3. The management staff will be able to read the whole table.

1.1.2.3.1. This will be outputted as a table on the system.

1.1.2.3.2. The system must allow records to be updated.

1.1.2.4. By the receptionists.

1.1.2.5. By the management staff.

1.1.2.6. Customers/guests should only be able to update their own record.

1.1.3. The system must allow the following stakeholders to update records:

1.1.3.1. Customer/guests.

1.1.3.2. The management staff.

1.1.3.3. The receptionists.

1.1.4. The system must allow records to be deleted by the following stakeholders:

1.1.4.1. Customer/guest. They should only be allowed to delete their own record.

1.1.4.2. Management staff.

1.1.4.3. Receptionist staff.

1.2. Booking:

1.2.1. The system must allow the creation of records:

1.2.1.1. The system must allow the following to stakeholders to create records:

1.2.1.1.1. The customer/guest. They should only be able to create a booking record that is linked to them.

1.2.1.1.2. The receptionist staff.

1.2.1.1.3. The management staff.

1.2.1.2. Validation:

1.2.1.2.1. ‘BookingID’ will be checked to make sure that the ‘BookingID’ entered does not already exist.

1.2.1.2.2. The ‘number_of_guests’ field will need to have a type check form of validation to check if it is an integer.

1.2.1.2.3. The ‘check_in’, ‘check_out’ and ‘date_of_booking’ fields will need to have a format check to check if they are in the correct format for a date.

1.2.1.2.4. The ‘RoomID’, ‘Type_of_HolidayID’ and ‘CustomerID’ foreign keys will need to be checked if they exist in their respective tables.
1.2.1.2.5. The ‘Offer_code’ field will need to be checked if it is a valid offer code by using a lookup check. There will be a list of valid offer codes.
1.2.1.2.6. A lookup check will be used on the ‘Holiday_Type’ field as there are only certain holidays the hotel will offer.
1.2.1.2.7. The ‘Price’ field should be validated using a type check to check if it is a floating point or not.

1.2.2. The system must allow the following stakeholders to read the booking table:
1.2.2.1. The customer/guest will only be allowed to read their own record.
1.2.2.1.1. This will be sent to the customer/guest via an automatic email one the booking is made.
1.2.2.1.2. If an offer code that is not valid an email should be sent, automatically, to the customer.
1.2.2.2. The management staff will be able to read the whole table.
1.2.2.2.1. This will be outputted as a table on the system.
1.2.2.2.2. The management staff will be able to view the bookings as a tape chart. This will be outputted by the system on- screen.
1.2.2.2.3. Management staff will be able to receive an output of number of bookings per month. This will be produced as a pie chart. This graph will be available to be outputted as a hard copy and on the system.
1.2.2.2.4. Management staff will also be able to view the number of bookings in a particular room. This will be displayed as a bar chart.This graph will be available to be outputted as a hard copy and on the system.
1.2.2.3. The receptionist will be able to read the whole table.
1.2.2.3.1. This will be outputted as a table on the system.
1.2.2.3.2. The receptionist will be able to view the bookings as a tape chart. This will be outputted by the system on-screen.

1.2.3. The system must allow the following stakeholders to update records:
1.2.3.1. Customer/guests. They must only be allowed to update their own booking record.
1.2.3.1.1. The updated record will must be checked if it clashes with any other dates.
1.2.3.2. The management staff.
1.2.3.3. The receptionists.

1.2.4. The system must allow the following stakeholders to delete records:
1.2.4.1. Customer/guest. They must only be allowed to delete their own records.

1.2.4.2. Management staff.
1.2.4.3. Receptionists.

1.3. Bill

1.3.1. The system must allow the creation of records:
1.3.1.1. The system must allow the creation of records by the following stakeholders:
1.3.1.1.1. The management staff.
1.3.1.1.2. The receptionist.
1.3.1.2. Validation:
1.3.1.2.1. ‘BillID’ will be checked to make sure that the ‘BillID’ entered does not already exist.
1.3.1.2.2. The fields ‘Mini_bar_bill’, ‘Main_bar_bill’, ‘Restaurant_bill’, ‘Telephone_bill’, ‘Spa_bill Room_service’, ‘Deposit_paid’, ‘Gym_fee’ and ‘Total_price’ will all have a type check as they all a currency and therefore have to be a float data type.
1.3.1.2.3. The foreign keys ‘BookingID’, ‘Payment_typeID’ and ‘CustomerID’ will be checked to see if they exist within their respective tables.
1.3.2. The system must allow the following stakeholders to read the bill table:
1.3.2.1. Customer/guest. They will only be able to read their own bill. This should be emailed to the customer as an attachment (attachment will be a word document).
1.3.2.2. Management staff. This will be outputted as a table in the system.
1.3.2.3. Receptionist staff. This will be outputted as a table in the system.
1.3.2.4. Accountants. This will be outputted as a table in the system.
1.3.3. The system must allow the following stakeholders to delete records:
1.3.3.1. Management staff.
1.3.3.2. Receptionist staff.
1.3.4. Accountants.

1.4. Room:

1.4.1. The system must all the creation of records:
1.4.1.1. The system must allow the following stakeholders to create records:
1.4.1.1.1. Management staff.
1.4.1.2. Validation:
1.4.1.2.1. ‘RoomID’ must be checked if it already exists within the table.

1.4.1.2.2. The fields ‘Room_Number’, ‘Price_of_room’, ‘Number_of_beds’, ‘Occupied’, ‘Maintained’ and ‘Floor’ will all use a data type as a form of validation.

1.4.1.2.2.1. The fields ‘Room_Number’, ‘Number_of_beds’ and ‘Floor’ will be checked to see if they are integers.

1.4.1.2.2.2. The field ‘Price_of_room’ will be checked to see if it is a float.

1.4.1.2.2.3. The field ‘Occupied’ will be checked to see if it is a boolean.

1.4.1.2.2.4. The field ‘maintained’ will be a boolean value and therefore have a type check.

1.4.1.2.3. ‘ViewID’ will have a lookup check as there are only a certain amount of views a hotel can have.

1.4.1.2.4. The foreign keys ‘Room_TypeID’, ‘Staff/RoomID’ and ‘FacilitiesID’’ will be checked to see if they exist in their respective tables.

1.4.2. The system must allow the following stakeholders to read the table:

1.4.2.1. Management staff. This should be outputted as a table on the system.

1.4.2.1.1. Management staff will also be able to view the number of bookings in a particular room. This will be displayed as a bar chart.This graph will be available to be outputted as a hard copy and on the system.

1.4.2.2. Customer/guest. This will be outputted as a list of rooms on the system.

1.4.2.3. Receptionists. This should be outputted as a table on the system.

1.4.3. The system must allow the following stakeholders to delete records:

1.4.3.1. Management staff.

1.5. Room_Type:

1.5.1. The system must allow the creation of records:

1.5.1.1. The system must allow the creation of records by the following stakeholders:

1.5.1.1.1. Management staff.

1.5.1.2. Validation:

1.5.1.2.1. The ‘Room_TypID’ field will need to be checked to see if it already exists as this is the primary key.

1.5.1.2.2. ‘Type_of_suite’ will be a presence check as something will need to be entered.

1.5.1.2.3. ‘Balcony’ and ‘Pets’ will be a type check as both of these fields will need to be boolean data type.

1.5.1.2.4. ‘Price’ will be a type check as this field will need to be a float.

1.5.2. The system must allow the following stakeholders to view records:

1.5.2.1. Management staff. This should be outputted as a table on the system.

1.5.3. The system must allow the following stakeholders to update records:

1.5.3.1. Management staff.

1.5.4. The system must allow the following stakeholders to delete records:

1.5.4.1. Management staff.

1.6. Facilities:

1.6.1. The system must allow the creation of records:

1.6.1.1. The system must allow the following stakeholders to create records:

1.6.1.1.1. Management staff.

1.6.1.2. Validation:

1.6.1.2.1. The ‘FacilitiesID’ field will need to be checked to see if it already exists as this is the primary key.

1.6.1.2.2. ‘Shower’, ‘Wi-Fi’, ‘Minibar’, ‘Safety deposit box’ and ‘Air-conditioning’ will all have a type check to check if they are boolean.

1.6.2. The system must allow the following stakeholders to read the table:

1.6.2.1. Management staff. This should be outputted as a table on the system.

1.6.3. The system must allow the following stakeholders to update records:

1.6.3.1. Management staff.

1.6.4. The system must allow the following stakeholders to delete the records:

1.6.4.1. Management staff.

1.7. Staff:

1.7.1. The system must allow the creation of records:

1.7.1.1. The system must allow the following stakeholders to create records:

1.7.1.1.1. Management staff.

1.7.1.2. Validation:

1.7.1.2.1. ‘StaffID’ will be checked to see if it already exists within the table.

1.7.1.2.2. ‘Job’ a lookup check will be used to simply check if the job exists within the hotel.

1.7.1.2.3. ‘Salary’ will be checked to see if it is a float.

1.7.1.2.4. ‘Frst_name’ and ‘Srn_name’ will be checked if they contain non-alpha-numeric characters.

1.7.1.2.5. ‘Contact_number’ will be a length check as a phone number can only be a certain number of characters long.

1.7.1.2.6. ‘DOB’,’Email’ and ‘Postcode’ will have a format check as both of these fields need to have a specific format.

1.7.1.2.7. ‘Username’ will be checked to see if it matches any other usernames in the system.

1.7.1.2.8. The ‘Password’ field will be checked if it is above a certain amount of characters, if it contains uppercase and lowercase characters and if it contains alphanumeric characters.

1.7.1.2.9. The ‘Access level’ field will have a presence check to see if anything has been entered as all employees should have an access level.

1.7.2. The following stakeholders should be able to read the table:

1.7.2.1. Management staff. They will view this as a table on the system.

1.7.2.2. Receptionists must be able to read their own record. This will be outputted through an email.

1.7.2.3. Cleaning staff must be able to read their own record. This will be outputted through an email.

1.7.2.4. Accountancy staff. They will view this as a table on the system.

1.7.2.5. Maintenance staff must be able to read their own record. This will be outputted through an email.

1.7.3. The following stakeholders will be able to update records:

1.7.3.1. Management staff will be able to update their own record.

1.7.3.2. Receptionists will be able to update their own record.

1.7.3.3. Cleaning staff will be able to update their own record.

1.7.3.4. Maintenance staff will be able to update their own record.

1.7.3.5. Accountancy staff.

1.7.4. The following stakeholders will be able to delete records:

1.7.4.1. Management staff.

2. Performance:

2.1. There will be 100% accuracy in calculations.

2.2. All inputs will take under 10 seconds.

2.3. All outputs should be received in under 10 seconds.

3. Usability:

3.1. My system will contain buttons so users will not need to type everything when using the system.

3.2. When a table is displayed the user will be able to select a record to update or delete records.

3.2.1. When a record is selected to update the entry fields will be filled with the data of the field to be updated.

3.3. My system must have tabs to switch through all the tables.

3.4. My system must have a separate window which opens when they wish to add or update a record.

3.5. When the user is searching for a record the record they are searching for must be highlighted.

3.5.1. If there is more than one record which meets the search requirements the system must create another table which will store this data.

4. Security:

4.1. Authentication provision:

4.1.1. Each user must have a username that is unique to them and they will also have a password.

4.1.2. The password must be checked if it contains numbers, upper case characters, lower case characters and special case characters (!,?,>,@).

4.2. Access levels:

4.2.1. Customers/guests will have access to their own records in the customer table, booking table and their bill table.

4.2.2. Management staff will have full read and write access for the whole database.

4.2.3. Accountancy staff will be able to read the:

4.2.3.1. Bill table.

4.2.3.2. Staff table (only the salary field). They will be able to read their own records on the staff table.

4.2.4. Receptionists will have read access to the following tables:

4.2.4.1. Customer table.

4.2.4.2. Booking table.

4.2.4.3. Room table.

4.2.4.4. Their own record in the staff table.

4.2.5. Receptionists will have the following write permissions:

4.2.5.1. Customer table.

4.2.5.2. Booking table.

4.2.5.3. Staff table. They will only be able to update their own record.

4.2.6. Maintenance and cleaning staff will have read access to the following tables:

4.2.6.1. Staff table (only their record in the staff table).

4.2.6.2. Room table.

4.3. Backup:

4.3.1. There will be an option for the manager to backup their system at any time by clicking a backup button.

4.3.2. My system must have a periodic backup. This will backup the database every month.

4.3.3. The backups will be stored in a file simply called ‘backup’ within the directory in which the script is held.

4.3.4. The system must delete the oldest backup when 4 backups are made to save space.

4.4. Encryption:

4.4.1. The ‘Credit_card_detials’ field in the customer table and the passwords of all the users of the system will be encrypted.

4.4.2. The ‘Credit_card_detials’ field in the customer table and the passwords of all the users of the system can be decrypted.