HOTEL RESERVATIONS SYSTEM

Final Year Project
B.Sc.(Hons) in Software Development

BY
NIALL MCCANN
G000344474

May 10, 2020

Advised by Dr. John Healy & Mark Campbell
Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

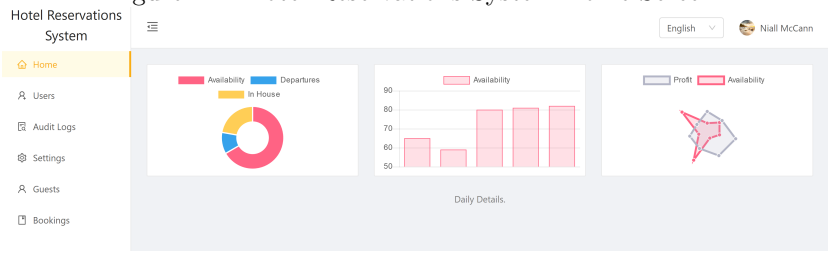
List of Figures

Chapter 1

Introduction

For my Final Year Level 8 Project, I have decided to base it on a Hotel Front Desk/Reservations System.

Figure 1.1: Hotel Reservations System Home Screen.



1.0.1 Background

Over the course of the past seven years I have been working at weekends and full-time during the summer as a hotel receptionist in The Abbeyglen Castle Hotel in Clifden, Connemara. From having worked as a receptionist for these years, and having experienced so many different situations within the hotel throughout these years, I have decided to base my project on this as I feel I have first hand experience with the good and the bad of these systems. When I first started in Abbeyglen, they used the system PAMS. This system had been discontinued since 1998, but was still in use in Abbeyglen daily. About a year into my work here, Brian, our manager decided to upgrade reception by introducing a new reservations system.

From his dealings with various other hotels he decided on buying Micros Opera for the hotel. There was so many benefits of upgrading to Micros Opera. With PAMS, we only had the system on one desktop, so if check out or check in was

in progress we were unable to input any bookings into the system. With PAMS being such an old system, that was no longer supported, meaning that it was unable to become integrated with the channel manager to allow bookings to input onto the system automatically. Micros Opera allowed us run our reservations side of the business so much more efficiently as multiple people could be using the system at the same time, working with data seamlessly. While having Micros Opera in the hotel for a number of years now, we are still learning and working hard into getting it completely set up.

Before Coronavirus had begun, we were hoping to finally have our channel manager integrated into the system by May, by unfortunately we are unsure of when that will all happen now. Having worked with PAMS and Micros Opera, I have a good knowledge of using these systems, and what I believe would be improvements and variations that could make them more user friendly and modern. From my own experience of these systems, and the collected experience of others working in this sector, through interviews and video calls, I really feel as though I have a good basis to create this project on.

1.0.2 Scope

The scope behind creating this project, is that from working very closely with these systems for several years, I really feel as though they could be improved immensely. What I'm really hoping to achieve from the creation of this project is a new, fresh feel reservations system, with a user friendly interface. The application should not be bulky, or cause the user to navigate through too many pop up windows just to create a booking. Everything the user could need to complete the reservations, should be clearly displayed on one screen where the booking can be entered into the system in the most efficient manor possible. The true purpose behind this project is to create a system that will save the users time, and let them use this spare time in a more efficient manor, instead of navigating pointlessly through unneeded windows.

Another feature I really want to implement into the project is to have it cloud based. From my research, most of the systems out there today are still network based, so I would really like to make my system run on the cloud so that the users would be able to work remotely should they ever have to. I know from in the Abbeyglen Castle, one of our workers lives on Inishbofin Island, off the west coast of Cleggan, and manages the majority of our telephone reservations. When we installed Micros Opera into the hotel we not only had to install in on the server within the hotel, but we needed to also install it on a separate server in her home, so that she would be able to access the database and be able to input reservations into the system.

From my own experience of these systems I find them extremely bulky, and the User Interface is very unpleasing and dated. From my research with various hotels, I have discovered many ways to make some improvements.

For my project, I have designed a Front Desk/Reservations System in React Native. With using React Native, I not only have learned a lot of new skills from learning a new language other than the various languages and skills that we have learned throughout our four years in Galway Mayo Institute of Technology, but have found it really interesting to see the capabilities of this language.
With my project, one of the key features I wanted was a new, easy to use interface. Had I written my project in Java (which most Hotel Front Desk Systems are written in), C, Python or many others, I would have found myself having a much heavier project, with a real unfriendly interface, that would have been very heavy for the computer to run, causing them to be much slower. With this, the project has a light fresh feel, and runs on the browser as opposed to many systems running locally on the server. A benefit of having it run on the browser is that it can be accessed anywhere, and at anytime, and in the difficult uncertain times we are experiencing at the moment with the global pandemic of Coronavirus, it is now more than ever that the importance of working from home has come into play.

1.0.3 Outline

An outline of this document is as follows :

Methodology :
In this chapter I give an insight into my approach of the Agile methodology which I used for coding and designing this application. I then proceed with my investigation into the problem, my planning behind how I was going to develop this application, my meetings with my supervisor, and then the development tools used throughout the process of development.

Technology Review :
For my technology review, this chapter describes my interviews with staff from various hotels throughout Connemara and Galway, and the different systems that I got experience of to aid in the designing of my own. I also give an outline of the difficulties of doing basic tasks on Micros Opera, and how my system compares to it. I then proceed to give a description of the various systems I experienced on my visits with all the different hotels. After this, I explain the benefits and why I decided to use Firebase, and also the different languages I was looking into using for coding the system.

System Design :
In this chapter I describe the findings of my interviews, and the following surveys that I sent to the different hotels I visited over the course of the year, and give the reasoning behind these features being in my final project, focusing on the need for a new clear to use interface, and the need for cloud hosting and storage for ease of access of the application and data.

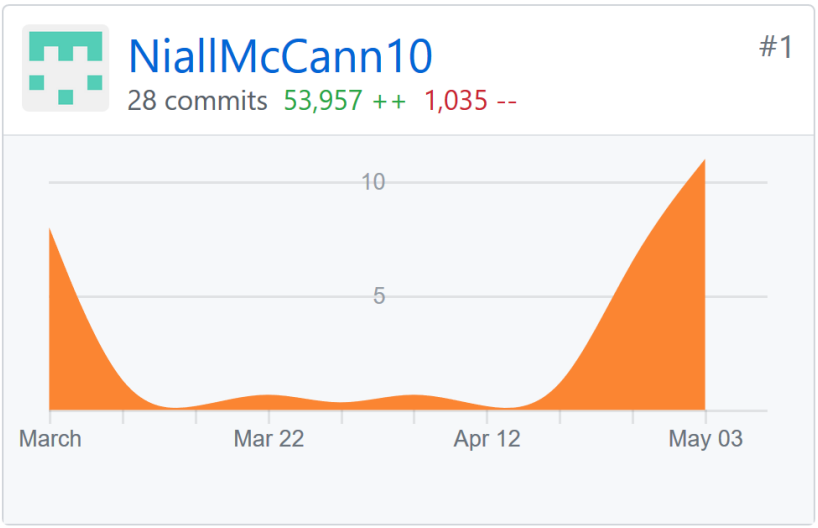
System Evaluation :
In this chapter I evaluate the system I created, and also give the results of my testing. For testing I gave the system to users in front of me and asked them to undertake a task, and where not possible to meet, I got my testers to screen share on Microsoft Teams and complete a task while talking me through what they felt of the system. In this chapter I also outline the limitations of the system.

Conclusion :
And finally in my conclusion I outline how I feel the end project turned out and the reviews I have received on it. I also review the great experience I got from creating this application, and the joy it was to visit so many different hotels. I then outline the opportunities that hopefully I will be able to complete sometime in the future, and finish on what I hope to use the system for and thank all the necessary parties.

1.0.4 Github

The project can be found here https://github.com/NiallMcCann10/Hotel-Front-Desk Stored on Github

Figure 1.2: GitHub Commit Graph.



In my Github repository, you will find my frontend file. This file contains all the runnings of the application that the user will interact with.

You will also find the backend file. This file contains all the running of the application that the users interactions with the frontend will trigger.

The .firebase file contains information that the server requires from the application to host the application correctly.

A copy of a PDF version of my Dissertation is stored on my Github, along with a PowerPoint presentation giving a quick introduction into my application, and a screen cast of the application in use.

Chapter 2

Methodology

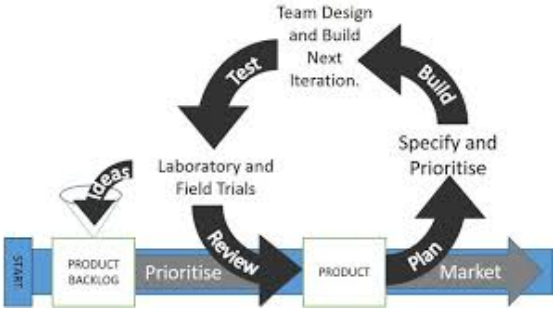
2.0.1 Introduction

In this chapter, I will explore the methodology I used in the designing of the application. Comparing Agile to Waterfall methodology. I then proceed to discuss my investigation and planning of the application. Then I explain the benefits of my regular meetings with Mark as a supervisor, and how I really benefited from these meetings. I then move on to describe the development tools behind the creation of this project.

2.0.2 Agile Definition

The approach I took to developing this project was of the Agile Methodology [1]. Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change. [2]

Figure 2.1: Agile Methodology Schematic.



In the beginning when I decided that I was going to undertake a project where I would be creating a hotel front desk/reservations system, I knew that this was an application that I would be working on by doing continuous iterations. Also with this project, I was continuously testing the various elements as I was creating them. The nice thing about using Agile methodology is that the creation of the project completely revolves around the customers needs. Although for this project I was the customer and the coder, I still felt this was the best methodology for this project.

One of the best parts of the Agile methodology is that with this methodology, you must be able to respond to change from the customer. What I really enjoyed with this was when my supervisor would give me an idea for a change, or where to go next, or if I asked one of my colleagues for their opinion, I was able to change my design for the better. Had I been using the Waterfall approach, then I would have stuck out with my initial ideas till the end, and my project may not have turned out as it has now.

2.0.3 Investigation

When it came to investigating the pros and cons of these systems, I felt I already had a good basis to start this project, and knew what I really wanted to achieve from the creation of this system from my own years of experience in the field. Initially I started out by asking my colleagues on their opinion of the Micros Opera system in which we use. I found this very beneficial as it turns out we all used the system quite differently. From these interviews alone, it changed my thoughts of Opera, and the ways in which I use the system, but also changed my views on how I wanted my final project to work. From here, I decided that to get a better understanding of these various systems that hotels use, and how different systems work in better or worse ways, that I would have to get experience of the first hand. So I decided to approach different hotels in my area, or hotels that I had a connection with, to get the front desk staffs opinions of how they use their systems, and what they think could be improved in them. Through contacting these hotels, I was able to meet with the staff using these systems daily and observe how they work on daily tasks, and get them to give me a tutorial into how to work these systems.

2.0.4 Planning

In September when I had decided that I would be designing a hotel front desk/reservations system, before I had started coding at all, I got in contact with numerous hotels around the area, most of which I had some connection into. I asked the staff if it would be possible for me to shadow them for a short while during the day, and get an insight into how they work the system that they use in these different hotels.

When it came to planning for the coding side of things, I spent a lot of time researching the best languages that I could write the project in. Initially looking at C and Java, which is the most common for these systems, but after extensive research I decided that I would like to try and learn something new for this project, so I decided to design this project using React Native, with Firebase as a hosting and database site.

2.0.5 Meetings

From my weekly meeting with our supervisor, I found them to be most beneficial. He was extremely helpful and encouraging when it came to any design ideas I had for the system. Our meetings were held every Monday, in first semester we would meet in a classroom just passed the canteen where our supervisor would talk with us each individually, to give us ideas of various projects to undertake, and then to check how we were progressing with the project and guide us in the right direction.

After Christmas, in the second semester, our supervisor would meet us again, every Monday in the canteen. In this time, he would guide us through the various checkpoints on the project to ensure we were meeting our deadlines. We would also give the supervisor a walk through of the project so far, and get his opinion on it. He was always extremely helpful and uplifting with anything he would have to say on your project. He would always make you feel motivated to undertake the next step, and would be brilliant in giving us ideas about where to progress next, and how to undertake the challenge that lay before us.

Our supervisor also gave me some great ideas about how to change my project, and my mindset behind it, which was brilliant to get somebody else views on how the system should be structured, and what someone else expects it to be capable of doing. He had also offered me to get in contact with one of his friends working in Aro Digital Strategies. Aro is a company that designs the booking system for hotel website, and the channel manager behind the booking engine. This would have been a great opportunity to be able to meet someone from industry and how they designed these systems, but I decided that since my project was just the reservations side, and not the channel management or website booking engine, that it wouldn't benefit my project, and focused on meetings with various hotels and their staff. Even now, with the unprecedented times that we are going through with COVID-19, our supervisor was still a pleasure to work with. He would have calls or messages with us weekly using Microsoft Teams, and provide us with useful information regarding the situation that we are going through, and also give us great feedback about the projects.

2.0.6 Development Tools

For the hosting of my application, I have all my code stored on Github. Github is an extremely useful way of storing your code, and being able to access it from different computers in different locations. For the database and deployment of my application, I decided to use the online storage and deployment system of Firebase.
Initially when I was researching into the ways in these systems are coded, it was very clear that most companies used either Java or C to develop them. so I started to create my project in Java. This all started out okay, but when it was coming to the user interface, I just wasn't happy with the aesthetics of the project. After all, the main purpose of the application was to create a new modern, easy to use interface. So I stopped coding with Java and started to research other languages that I could code the project in. Initially, I was drawn to Ionic and Angular, these gave a great base to start the project on, and it would have a really nice interface. This would have been great, but then I saw React Native. Its extremely similar to Ionic and Angular, gives a really fresh and modern feel, but at the same time, I was learning a new language to code for my final year project.

2.0.7 Summary

In summary of this chapter, I discussed the Agile methodology, and the benefits of using this for the development of my application. I also discuss my investigation and planning processes behind the development of this application, and how I really benefited from my regular meetings with Mark. Then to conclude this chapter I cover the development tools used in the creation of this project.

Chapter 3

Technology Review

3.0.1 Introduction

In this chapter, I review the numerous Micros Opera in more depth, and evaluate each of the different systems that the various hotels I visited used. I give an analysis of why I chose Firebase as my database and hosting site, and why I decided to write the program in React Native.

3.0.2 Technology Review

For my Technology Review, I travelled to various hotels around Galway City and County and further. Such as The Ballynahinch Castle Hotel, Clifden Station House Hotel, The Forster Court Hotel, Renvyle House Hotel, The Menlo Park Hotel and also Lough Rynn Castle Hotel. While meeting with Management/Front desk staff of these hotels, I discovered many features about the various systems used that people dislike or ways in which they would like to change the systems, I also learned of many ways in which they love each of their systems, and the functionality they provide.
From my meetings with the various hotels, I found that one system was a true winner in popularity. HotSoft, clearly being the most popular with Ballynahinch Castle, Clifden Station House and Forster Court all using it. With Hotsoft, it has a truly modern Interface with a friendly, easy to use feel. Hotsoft being tailor made for your hotel provides so many benefits, such as special room types that smaller boutique hotels may have, special charges that may be applied on billing, or even having the functionality to have the restaurant booking engine built in. This was the only system from all that I studied that was able to provide this functionality.
Being a cloud based program, this meant that it was easily accessible across many different devices, with no fear of any data loss should a computer crash. Another advantage of this being that it is possible to work from home or on holiday once you have access to the internet.

With my own experience with Pams and Opera, and with the collected data from these various hotels, I was able to make a strong argument about why these changes should take place, and how different people and hotels use these systems.
Throughout my research I also discovered that a lot of various companies are willing to give you a free sample of their software to test. I found this most useful as not only being able to experience the software by visiting the hotels, I was also able to download my own version and discover my own thoughts towards them.

Having researched these various software's, I discovered that the majority of times people want their software to be fully integrated with all the different booking channels. Throughout my visits with these hotels, I discovered that hotels deal with channels in many different ways. In the Abbeyglen Castle Hotel, Lough Rynn Castle and Ballynahinch Castle Hotel, the booking channels are controlled by the chain they are managed by, Original Irish Hotels of Ireland.

But in hotels like Renvyle House and The Clifden Station House Hotel, being smaller independent hotels, they would manage each booking channel separately and by their own reservations team. With this being said, it makes it so important in today's world, where more and more people are booking hotels through various websites, that all channels are fully integrated and reservations and availability are easily managed to avoid over bookings and mistakes.

With Opera, myself and the team in Abbeyglen have found it to be a good system overall, but could definitely be modernised and made somewhat more efficient and user friendly.
Initially for logging into the Opera system, the interface isn't that striking. In the top left hand side you enter your username and password, where it will then prompt you to select "PMS" to proceed to the application. With Opera, your password must be changed monthly for security purposes, and with that you are unable to repeat a previous password until 4 months has passed since its use.
Once in the main application, you're presented with the main menu screen. On this screen the main selections are presented along the top of the screen, where the sub selections are displayed down the left hand side. To select a menu choice, you must double click onto the button. Once selected the screen will bring you to the selected menu with sub selections.

3.0.3 Opera New Reservation

In our "Reservations" heading, we are able to choose in the sub selections to create a new reservation.

To create a new reservation in Opera, you must double click onto the "New Reservation" tab. This will then create a pop up window, prompting the user to enter the arrival date, number of nights, number of guests, and number of rooms. The user then enters the guests name and selects "OK".
From here the user is then brought to another pop up window where they will search the database to establish if the guest has stayed with us before. If so, the guests information, along with when they last stayed with us will appear in the bottom grid. If the guest has stayed previously, then we select the guest information from the grid and select "OK". If the guest hasn't stayed before then we will select "New".

From there another new pop up window will appear, prompting the user to enter more information about the guest. In this window this is where we would enter the guests address, contact details and nationality. Once this is complete the user must then select "OK".

The user is then brought to another pop up window again, where they must select the rate package and the room type the guest is booking. Along the top is the selections of rooms, ranging from Standard, Suite, Superior to Economy. Along the left side of the screen is the rate packages from Bed and Breakfast, Dinner, Bed and Breakfast, complimentary and dinner on various nights included. Once the user selects the room type and rate package from the grid below, they again select "OK".

The user is then brought to the main screen of the reservation, where much of the information they inputted previously is displayed, from the name and contact details, up the top left, dropping to the arrival date and the amount of nights, guests and rooms. Now underneath the user must input the nightly rate for the guests stay and the room allocation. If the user has chosen a Superior room in the previous window, but allocates a room number that is registered as a different type, then Opera will alert the user to this error, and prompt them to input a correct room number for the appropriate room type.

The user will then input the destination of which the booking came from, eg. Direct booking, Original Irish Hotels booking, Central Reservations, or various online sources.

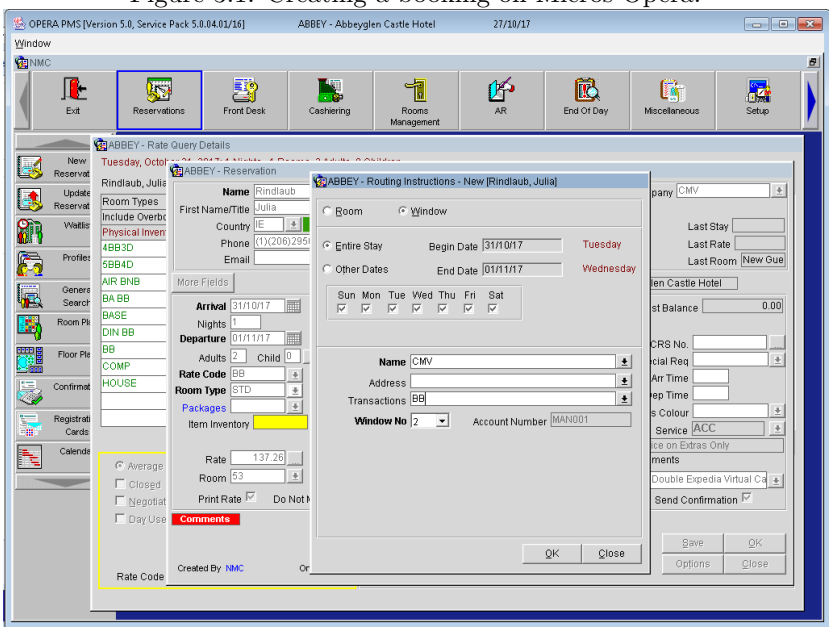
Under that, the user must then input the comments for the reservation. This is where the user specifies if the guest would like a double or twin room, which package the reservation has attached to it, and the complete rate for the stay.This is also where the user will input if the guest has any allergies, or

special dietary needs, or should the guest have any other special requests or birthday/anniversary notifications.

If the reservation is to be paid directly to the hotel, then the user can select "save" then "OK" and the reservation is finally complete. But with so many bookings being made and paid for through online agencies now, the hotel must invoice the various companies for the rate for the stay of the guest in the hotel, so in doing that we are splitting the bill, so that on one side there is the extras that the guest will owe the hotel directly, where on the other side will be the accommodation charge which the hotel will be invoicing.

To make this possible, before clicking "save", the user must select "options". This will make a new pop up appear where the user can select numerous options, the most popular being Alerts, Confirmation, Deposits and Routing. To create the separate bills the user must select "Routing". Once selected, again the user will be presented with another pop up window. In this pop up, the user selects "window" up the top, the clicks the drop down arrow beside the guests name, and they will be able to select the appropriate company to invoice for the stay. In the transactions drop down, the user can select the rate coding to be invoiced for, eg. Bed and Breakfast, or Dinner, Bed and Breakfast.

Figure 3.1: Creating a booking on Micros Opera.

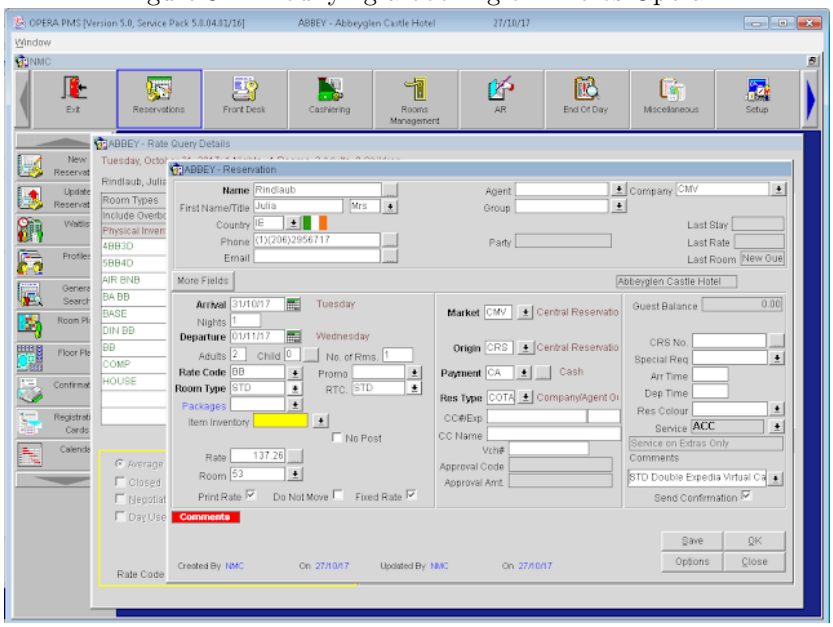


After selecting "OK", the user can final save and finish creating the booking.

3.0.4 Opera Modify Reservation

To modify a reservation, when in the main menu for the reservations tab, the user selects "Update Reservation"
From there the user must enter the name and arrival date of the guest to locate the reservation. Once the reservation is located the user can modify or cancel the booking from here.

Figure 3.2: Modifying a booking on Micros Opera.



3.0.5 Opera Check In

To check in a guest, the user must select "Front Desk" from the top menu bar, from here they must then select "Arrivals". The user is then brought to a pop up window, where they will have to select the arrival date they desire, and click search.
Once the arrivals appear, the user must then select the desired guest, confirm their details are correct, take the guests credit or debit card details and then they can select "Check In". This will then mark the guest as an in-house guest until they check out.

3.0.6 HotSoft

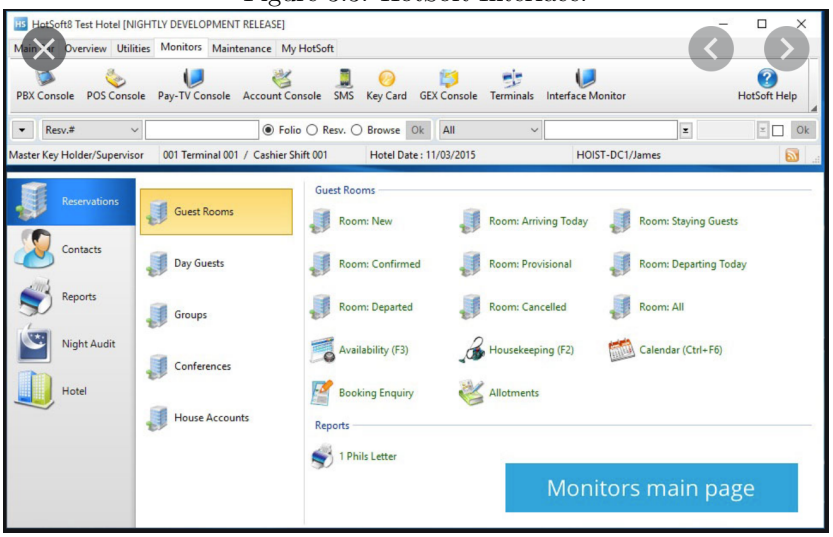
With Hotsoft, which from my findings being the most popular, these systems are fully integrated with the channel managers, but with Micros Opera, Visual One, and Room Master, these systems are older and much harder to integrate. [3]

Unfortunately from my findings, Hotsoft isn't cloud based, which many find disappointing. One of the reviews even stated that with the difficult unprecedented times that we are currently going through, if the system had been cloud based it would make it so much easier for her to work from home, where she would still be able to retain employment, instead of having lost her position due to COVID-19.

But Hotsoft was still the most popular being in use in Clifden Station House, Forster Court, and Ballynahinch Castle. Although it may be let down by the fact that its not a cloud based system, it still has many positives which out way the negatives. This system is tailor built for your hotel, making it unique to each location. It comes with full housekeeping integration, channel management integration, and has the ability to create gift vouchers and manage the amount present on the vouchers by a simple swipe of the card.

Each hotel sang the praises of Hotsoft saying that it has made their jobs a lot easier and more efficient, compared to when they were running their previous software's.

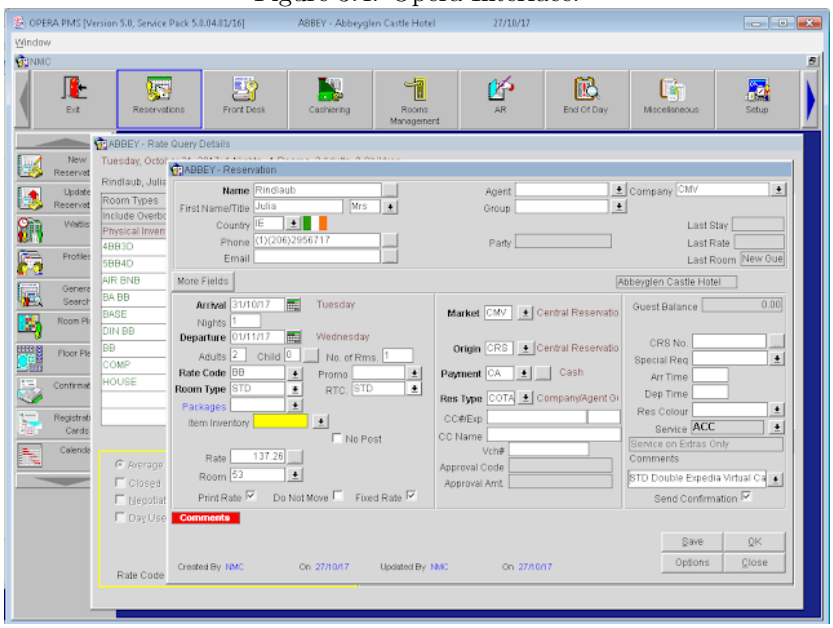
Figure 3.3: HotSoft Interface.



3.0.7 Micros Opera

Although Micros Opera is an older system, with a much bulkier, less user friendly interface [4], and less popular with hotels (Clifden Station House and Ballynahinch Castle both changed to Hotsoft), it is still a highly thought of system for hotels to use.
Although with Micros Opera just to make a reservation you are required to fill in details on several separate pop up windows, this making it extremely heavy to use, but also difficult for staff to input reservations at a quick simple pace. A plus to Micros Opera is that with a simple click of a button a user is able to check availability and room allocations. These may not be the clearest symbols, or the most user friendly windows, but they are simple to access, which is massively important for the efficiency of the running of the hotel.
Micros Opera may be somewhat slow compared to Hotsoft, but it does still have its advantages. Opera is a very robust system, which doesn't crash easily. It can be running on numerous machines all accessing the database, but will always keep the most recent data available to each user.
Micros Opera is also able to be made tailored to each hotel. This makes using the system easier for the users, but unfortunately the system is unable to be used to value gift vouchers, or have housekeeping integration wirelessly. The housekeepers would have to log into the system on a desktop and mark each room as clean from there, not on a mobile device that they could carry freely from room to room.

Figure 3.4: Opera Interface.



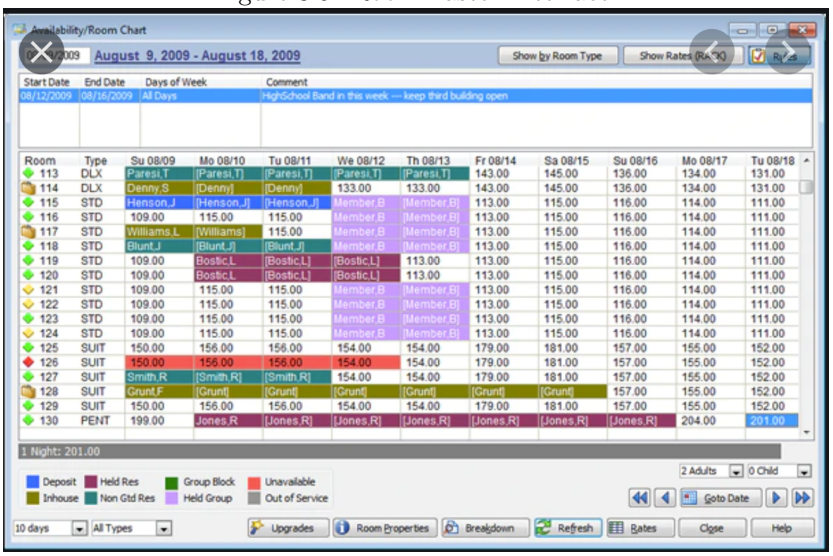
3.0.8 RoomMaster

Room Master, from my interview with Caroline in Renvyle House, is an overly complicated system.

Simple tasks, such as allocations, and availability, that have to be undertaken regularly by receptionists require many clicks, and screen changes just to check. The dated interface is unfriendly to the user, and system regularly crashes during the check out, when the system is in most demand

When Caroline looked further into the system, she discovered that the version that Renvyle House are using is an extremely outdated version that was produced in 2001. From researching this system online, it seems that they have made many improvements and have done a great deal to modernise the system and make it much more user friendly. [5]

Figure 3.5: RoomMaster Interface.



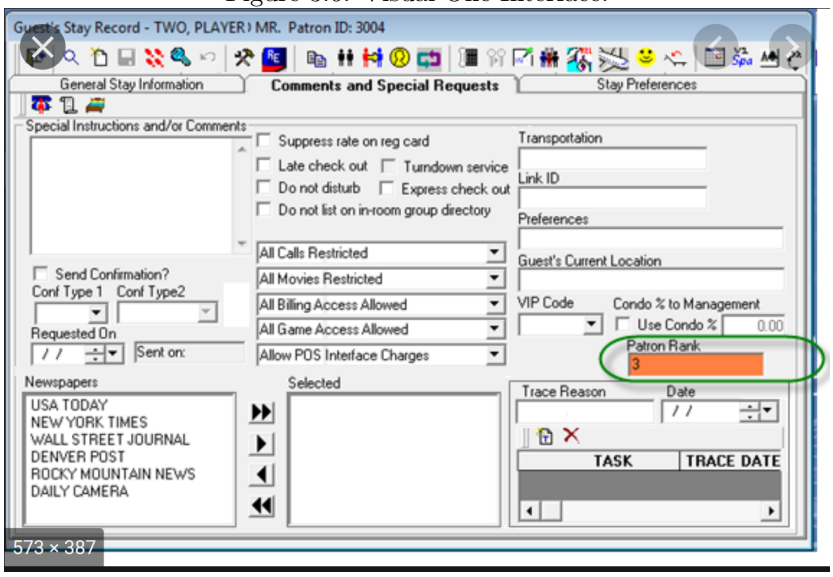
3.0.9 Visual One

Visual One which is an extremely efficient reservations/front desk system, but has an old style interface that isn't appealing to the users of the 21st century.

The interface may not be the nicest, but it has a very simple layout with every details for each section on the one window, making it easy for staff to input booking, check in/out guests, or manage availability.

From my interviews with Kevin in Lough Rynn Castle, he found that the system could be updated massively. The complete interface is outdated, with bad symbols, and in need of better labelling for the user to understand. Kevin also mentioned that unfortunately the system is unable to integrate with the channel manager for online bookings, so in the near future Lough Rynn would be looking to upgrade to a new system. From my findings of this project, I let Kevin know my own opinions of the different systems I have studied and shared my findings from the interviews to help Lough Rynn choose the right system for them. [6]

Figure 3.6: Visual One Interface.

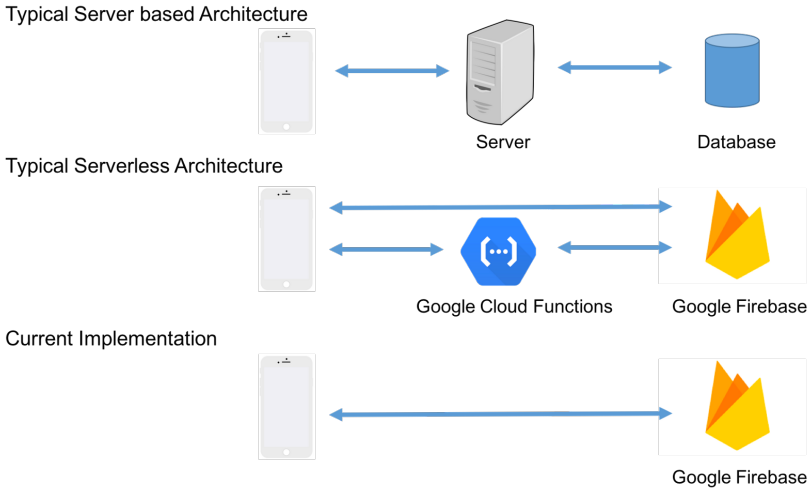


3.0.10 Database/Hosting

For the database and hosting of this application, I decided to use Firebase. There are many advantages to using Firebase

- **Storage** : Firebase storage feature is powered by Google Cloud Storage and allows users to easily download media files and visual contents. This feature is also helpful in making use of user-generated content.
- **Authentication and Hosting** : Firebase backend service offers a powerful authentication feature. It comes equipped with simple SDKs and easy to use libraries to integrate authentication feature with any mobile app.
- **Serverless Applications** : A benefit of firebase is that its a serverless application. This requires less onsite support and technology, and should there ever be a problem, the cloud will have a recent back up of all your data. A true benefit is that this data can be accessed anywhere with the right credentials, making working remotely much easier.

Figure 3.7: Firebase server.



- **Remote Configuration** : This feature in Firebase allows developers to incorporate changes in the app remotely. Thanks to this, the changes are reflected in the existing version, and the user does not need to download the latest updated version of the application. Making running and using the app much more pleasing for the user.
- **Performance Monitoring** : This feature brings automation to performance tracking of your app in real-time. This tool provides performance insights based on the root cause, tracing not only the app performance, but

also server connection quality and response time across different network types.

- **Configurations and Improvements** : The error reporting can also be configured remotely to control who can access the reports and list of events that occurred before an event.
- **Free** : Crash and bug reporting is free with Firebase. You don't need to pay anything to avail of this feature.
- **Email Alerts** : It allows sending email alerts as and when such issues or problems are detected. It also makes sending notification emails to users for verification extremely easy. [7]

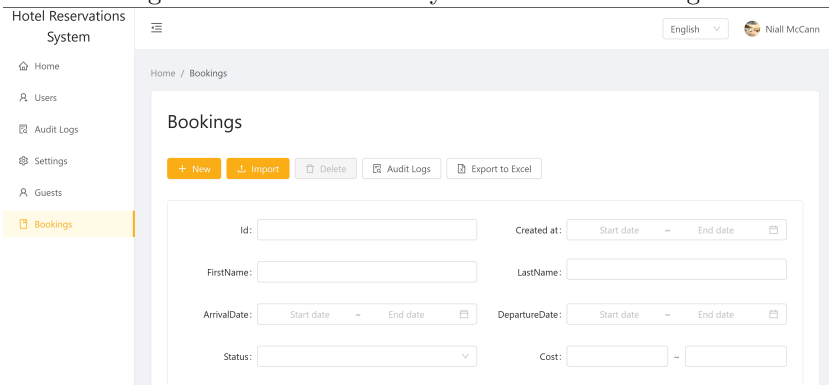
Before deciding to use Firebase, I had initially looked at using MySQL, as I had a lot of experience using this database from previous modules that we have undertaken throughout the four years in college. But after my research into how well Firebase works, and how simple it is to configure and integrate into React Native, I decided that it was the best option for me. Not only was it a great system to use for my database and for hosting the webpage, but it was also a new technology that I hadn't used previously.

3.0.11 Languages

For my search in the best language to undertake this project in, I decided to work with React Native. [8]
From my research of React Native before I began coding, I was already intrigued by it. React Native had been called a "real asset when it comes to improving the performances through native controls and modules". React Native connects the components for both the Operating Systems and generates a code to the native API's. A big advantage of React Native is the fact that you can reuse the code and the pre-developed components. With this feature in mind, it saves the developer a lot of time developing the application. A benefit that I really found interesting and useful was the fact that while using React Native, you can reuse the web applications code for using it on a mobile device.
A real seller for using React Native for me was the modern simple user interface that it offers. Based around many mobile apps that the general public are so familiar with these days, it makes daily use, and training for new employees so much easier.
This was one of the main details I wanted in my project, and feel that I have really captured this friendly interface through using React Native. As shown in the image below Fig.3.8, this is my system for creating a reservation, and I really feel as though that modern, fresh and simple interface has been captured through this.

What I found most beneficial about using React Native was the large community that surrounds it. Should I have run into trouble, or needed help figuring

Figure 3.8: Reservations System create a booking.



out how to undertake a problem, I could easily gain the information I needed through videos or online forums.

Before I had decided on using React Native, I had attempted to start my project using Python. The reasoning for choosing Python was because I really enjoyed using this language when we coded various projects for Ian in the past. Python is also a language that is very simple to get an understanding of.
But after some research, I decided that python wouldn't be the right choice for this application. Python has very limited database access. When compared with the more popular languages like C, Java or Ionic, Pythons database access layer is said to be underdeveloped. Another reason I decided against Python was that it isn't a very language to use for mobile development. With so many new hotel software's, they are needed to be mobile, that employees can use them on the go, wherever they may be based. This is one of the primary reasons I decided to use React Native for this project. Its a fast, reliable and mobile friendly development language that can be used across a multitude of devices.

I had also looked into using Java or C for programming this application, but decided that they wouldn't be right for the project, as they wouldn't allow me create the same modern interface as React Native has, and they don't have a lot of the useful built in icons and features that comes along with React Native. Also when it comes to working with Firebase, these more modern languages integrate with it so much easier than C or Java.

3.0.12 Summary

To summarise this chapter, I started with giving a summary of the how I discovered different hotel technologies by travelling to various hotels throughout Galway. I then proceed to describe the difficulties involved in undertaking the most basic of tasks in Micros Opera, and give a brief description of each of the systems the hotels from my interviews used. I then outline the benefits of

Firebase and why I decided to use this software for storage and hosting of my application. I then move on to summaries why I decided to use React Native, and the different languages I studied before settling on React Native.

Chapter 4

System Design

4.0.1 Introduction

In this chapter, I talk through the interviews I had with different hotels and also the results of my following survey of the improvements that each user thought they could benefit from most. I proceed to the to outline the main design features I wanted to include in my project.

For my system design, I feel I had a good basis for what I really wanted to create. What I wanted to achieve with this project was to create an easy to use interface for the user, but also complete bookings and reservations in a simple, easy to explain way.

4.0.2 Interviews

With my own work in a hotel reception for several years, and having used two separate systems in this time, I felt I had good reasoning behind my changes.

Not only from my own opinion did I get the ideas behind this project, but also from having talked to my colleagues in The Abbeyglen Castle Hotel, and from talking with staff in Ballynahinch Castle Hotel, Clifden Station House Hotel, Renvyle House Hotel, The Forster Court Hotel, Lough Rynn Castle Hotel and The Menlo Park Hotel.

From my conversations with these various people working in industry, I gained great experience behind different software systems used by hotels, but also what people loved and hated about these systems.

The staff graciously allowed me to shadow them, and ask questions about the functionality of the system. I found this first hand information brilliant for creating my competitor analysis and for the creation of my overall application.

After my visits to these hotels, I sent the workers a quick survey using Microsoft forms. In this survey, they detailed of how easy or difficult they found many of the different processes of using their system, but also what I found most beneficial was at the end they wrote a short paragraph about what they most felt would make improvements in the systems that they use daily. Fig.4.1

Figure 4.1: Results of survey on improvements.

11. What in your opinion would improve your system

6 Responses

ID ↑	Name	Responses
1	anonymous	A recovery feature to recover lost work of the system shuts you out
2	anonymous	Make availability clearer, not having to assign rooms when they make the booking.
3	anonymous	Integration would make it easier
4	anonymous	Daily use is easy and very easy to train someone in, but it can be hard to see future booking. Such as to run a report showing only our corporate arrivals for the next 2 weeks in a handy little printout.
5	anonymous	If it was cloud based, it would be easier to work from home. Also a better module for activity reservations & a linked restaurant reservation system so all parts of a reservation can be completed from one system.
6	anonymous	Clearer Labelling/Symbols and a clearer layout

These results clearly state that employees are really looking for a much simpler interface. Stating that they would enjoy the workings of the system more should it have "Clearer labelling/symbols and a clearer layout".

Another problem that staff find is that with it not being cloud based, that the lose data when the systems crashes, or that they are now out of work as they cannot work from home on the system.

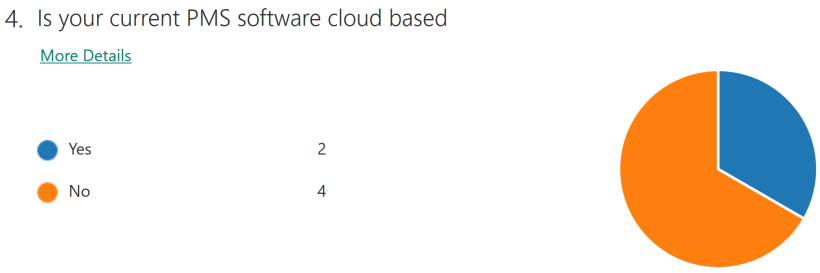
4.0.3 User Interface

The user interface is an important part of any application, and if the users don't like the way the system works, that can make or break any piece of software. I really wanted to create a simple, user friendly interface for my application. From my own experience, and the collected data from my interviews and surveys, it appears that nearly everyone I spoke with found the same.
Kevin in Lough Rynn would love clearer labelling and symbols in the system making it easier to understand and to train staff in. Catherine, from her experience with many systems over the years found Hotsoft to have an extremely friendly interface compared to competitors that she has used, but still feels as though there is room for improvement in the looks of these systems. With using React Native, I was able to create a completely new feeling interface, that is simple to navigate, and users feel familiar with, as it is very similar to many of the websites that we interact with daily.

4.0.4 Cloud Hosting

One of the main factors I wanted to make for my project, was that it would be cloud based. Not only do I find it a restriction having the system based on the server, but all of my interviews find the same.

Figure 4.2: Results of survey on Cloud based systems.



One of the main advantages of having the system cloud based [9], is that it cuts the cost of storage to a fraction of the cost of a server based software, but it also removes the need to train a member of staff to manage this equipment, which commonly gets outsourced for a high price.

Another advantage of your software being cloud based is that you don't have to waste any time or money on installation, its all just there when you need it and stored on the cloud. Accessible whenever you need it. Not only does this make the system faster, it also makes your data more secure. Once your data is stored in the cloud, it's easier to get a back-up or recovery of that data, which is an extremely long process on-site.

A benefit of cloud computing that so many people are finding in these difficult times, is the ability to work from home. Without cloud computing more and more people would be unable to work from home, making unemployment levels even higher, and costing the state more.

4.0.5 Summary

To summarise this chapter, we discuss the main aspects of the system that I would like to make through the creation of this system. We parse the information that I gathered from my surveys. I explore the main problems I hope to overcome with application, and my own opinion and the opinion of others in industry of these features, mainly user interface, and cloud computing.

Chapter 5

System Evaluation

5.0.1 Introduction

My overall evaluation of the project is positive. I am pleased with the final outcome and how I undertook the project. The project covers all the specifications that I was hoping to achieve by creating this system, and in this chapter I will discuss the method and outcome of testing, and the limitations of the system developed.

5.0.2 Testing

From my testing of the application, the feedback I received was positive. My approach to testing came in two different forms. I had asked some of my colleagues and family to undertake a simple task on the application and talk their way through it either in front of me, or via video call through Microsoft teams using screen sharing, using the Think Aloud methodology [10]. For the purposes of testing the application I had ten different people attempt to undertake different tasks on the system for me.

On one of my calls with our supervisor, he ran the application on his device, created an account and proceeded to modify a reservation with ease. He talked his way through the whole workings of signing in, and modifying the reservation, and commented that the only thing he would like to see changed is the pictures during the signup process. The pictures I had initially used were of low resolution and became pixelated, so with my supervisors comments in mind, I decided to use photos I had taken myself of the Connemara Region, which keep the project screen general, rather than based on one hotel, and incorporates in one of my other hobbies to the project.

Grainne in Abbeyglen found the project to be " very inviting". That from the moment she logged on, she loved the feel of the interface and how easily you can navigate from different sections. I asked Grainne to create a booking

on the system. She managed this with ease, and was impressed with how simple and how few windows you had to input information into to create a booking.

Kevin in Lough Rynn loved the fact that you can make it more personal. He really enjoyed being able to make the background colours to his choice, and being able to upload a profile picture gave it a much more personal touch. I asked Kevin to create a new guest profile in the system. He managed to navigate throughout the entire process with no trouble at all. Creating the new guest profile and stating that it was so aesthetically pleasing compared to using Visual Ones outdated interface.

I asked my mother, who has no experience of using hotel software whatsoever to create an account, and to delete a guests profile from the system. She managed to do so with no help from me and claimed that the system was very well laid out and self explanatory. She was able to undertake all these exercise by herself and was happy to trying out other features of the application.

My brother Alan, who has a degree in Computer Science, also tested the application for me. Having never worked in a hotel, but having more than ten years experience in the software industry, he was a hard critic. He thought the system was good, but that I could have spent longer creating more functionality, rather than providing the options for customisation. I explained how I really wanted to create a new, user friendly interface as one of the main aspects of my project, and showed him the findings of my interviews, and how tedious it is to create a booking on Micros Opera, and he soon understood my decision for the customisation.

Ronan in the Clifden Station House tested my system, and was impressed with its ease of use. I gave Ronan the task to create an account. Once his account was created I, as a manager gave him Manager credentials also. This gave Ronan more functionality over the system than just a newly registered user. Once he had this higher role, I asked Ronan to change the role of another user. Watching Ronan using the screen in front of me, he managed to find the users tab, navigate to a user who only had a basic role in the system, and edit the users profile to upgrade them to employee status with no difficulty whatsoever. Having been impressed with the ease of use, he called out his colleague Hannah to attempt to undertake a task.

I asked Hannah to create a new reservation for a guest that hasn't already got a profile in the system. She discovered that it isn't necessary to go to guests first, create a new profile, and then create the booking, she was very happy to discover that you can create a new guest profile during the process of creating a booking, that it will simple give you one pop up window to enter all the necessary guests details

My friend Ciaran, who is studying Construction Management in Galway Mayo

Institute of Technology also tested my system for me. I found it interesting to get someone outside of the hotel and technology sector to try my application, as this would test its usability for all types of people. I asked Ciaran to create an account, then to immediately sign out again, and attempt to reset his password. Through watching Ciaran use the system over a Microsoft Teams video, it was interesting to see his approach to how to undergo this task. After creating an account and was brought to the home page, it was interesting to see that to try and log out, Ciaran clicked into the "settings" tab. After discovering that this is only to change the colours of the selected items, he then changed his tactics and found the correct way to log out by clicking his name in the top right hand corner, and moving down the drop down list to sign out function. After he successfully logged out, he clicked onto the "forgot password" link, which brought him to the reset password page. He entered his email, and a new password was sent to his email for verification. Ciaran overall found the application very simple to navigate, even though he ran into some trouble in the beginning, was quickly able to fix his mistakes.

Caroline from Renvyle House also tested my application, I asked Caroline to create an account on the system, and then export her bookings for the next week to excel, so that she would be able to see her incoming guests details for the upcoming week. Caroline managed with ease to create her account, and navigate to the bookings tab. After having a look through the upcoming bookings, Caroline soon was able to select the arrival to and from dates from the input options above, and restrict the bookings she saw, to the selected dates. From here she very simple clicked the "Export to Excel" button above the selection options, and could download the excel spreadsheet with the reservations desired on it.

Lastly, I asked Catherine in Ballynahinch Castle, who has the longest experience working in hotels out of all my interviews, to test every aspect of the system. She created an account, changed her password, created a guests profile, created a booking, modified the booking, and deleted a booking. Changed the users roles, exported her guests details to excel, and uploaded a profile picture, and changed the colour of the options to her own choosing. Catherine said that she was very impressed with the system so far, and would love to see it taken further to complete all the functionality that a reservations/front desk would need to complete all their daily workload. She hopes to see it complete in the coming years, and that it will be available to but for hotel use in the future.

So in conclusion of my testing, the feedback I received was mostly of a positive style. From the few changes or recommendations that I received I graciously accepted them, and put them to use. Marks comments about my blurred images was correct, and after searching the internet, I soon found high quality images to replace my old pixelated images with. Overall everyone seemed to love the interface and how easy it is to use the system. This has given me a great belief in the system, and hope to develop it more to hopefully be able to sell the

application in the future.

5.0.3 Limitations

A limitation of using firebase as my hosting site is that if the application is ever to go into mass use, that I fear firebase wouldn't be able to cope with a mass amount of users attempting to access the same website at the same time. It may crash as from reading various reports online, Firebase is only capable of having 50 users accessing one system at a given time. This would be fine for the majority of hotels, as they wouldn't have massive teams trying to access the system all at once, but for a chain of hotels with a central reservations office, where a large team of staff would accessing and inputting bookings, or working on the system in some other way, may cause problems and the system may not be able cope with the pressure.

Other limitations include what I hope to add to the application in the future, which is the creation of Room Allocations, Built in billing functionality, Housekeeping integration, and once the system has an easier to user check in function, to create a guest self check in function to remove the need for a permanent receptionist on duty.

5.0.4 Summary

In this chapter, I evaluate my system from my own perspective, but also from my testing methods of my family members, friends and of personnel in industry. I then proceed to discuss some of the limitations of my system, and how I would like to progress the system.

Chapter 6

Conclusion

In conclusion I am very happy with the way that my project turned out. My objectives were to create a hotel front desk/reservations system that would be user friendly, with a modern interface, and be quick and easy to use. Overall I am extremely pleased with how my project turned out. The user interface has a fresh crisp, and modern feel, that can be customised by each user by changing the options colour, and by adding a profile picture. From my experience of other systems in hotels, there are many different windows the user must enter details into, just to create a booking, whereas in my project, I am very happy to say, all the information that is needed to be inputted by the user is clearly visible one the one page. This not only makes the system faster for the user, but it also saves so much time that the user could be focusing on other work with. I am also very pleased that my project runs from the cloud. This not only saves time installing the software, but it makes it so much easier for the employees to work remotely. This also eliminates the potential data loss of a server crashing while in use, making the data more secure. The benefit of this is that should one online server go down, then another "slave" will take over becoming the new "master" and keep the system running with its back up of all the database.

6.0.1 Experience

The experience of building this final year project was one I really enjoyed. When we were deciding for project that we would like to undertake at the beginning of the year, Mark told us choose a project that you would have an interest in. I truly found this project interesting as I really enjoy my work as a hotel receptionist, and having such experience in this field made the research and the ideas behind the changed to make to these systems so much easier. One of the best experiences I had from this whole project was the pleasure I had of visiting so many beautiful different hotels, and getting an insight into how they run daily, and how they use their front desk system. I found it extremely interesting to see all the different systems used, and how they differ. There was one clear winner for what seemed to be the best system, and that was definitely

HotSoft. The system was truly state of the art, and had really thought of everything a receptionist or reservations team would possibly need on the job. This project wasn't only useful to cover my final year project, but it was brilliant for my own working as a receptionist. I discovered so many new ways in which to use our system more efficiently, and really use it to its full ability, but I was also able to report back to my manager about my findings, and give him useful insights into what each hotel thought of the system they use, and what systems was the overall favourite. From here he gained crucial information first hand for upgrading our own working environment in Abbeyglen. From this he also got me to give advice to all the reception and reservations staff in Abbeyglen about how other hotels are running, and how we could make improvements in our daily work to run more efficiently and create a better working system for us all.
I really enjoyed my meetings with Mark too. Mark was an absolute pleasure to work with throughout the year. He was always extremely encouraging and uplifting with any ideas that we had for our projects. He was always available to give advice, or try his best to get information on a topic or assistance from someone he knew to help with our problems.

6.0.2 Opportunities

Going on with this project in the future, there would be many opportunities that you could undertake to make this project better.

- **Room Allocations** :
  Once a hotel decides to use this system, we would have to create the room allocations list that would be specific to each hotel. This would entail creating a limitation of how many rooms that this specific hotel has to let, but also with the specific room types that each hotel would have, eg. Standard, superior, suite, family room, apartment and so on.

- **Built in Billing** :
  Another improvement that could be added to the system would be to build in a billing function should this system be used for the whole reception, rather than just reservations. At the moment I have the application created so that the user can upload a copy of the bill to the users account so that should there be any discrepancies in the bill in the future, the user can search the guests profile, and the bill will be there. But if I was developing this system further, I would definitely like to add the functionality to have an inbuilt billing choice, where any extras can be charged to the room directly, rather than to a separate file.

- **Housekeeping** :
  A feature which would be extremely useful on the system would be to get housekeeping integrated into the system. With this the housekeepers would be able to mark each room clean on the system as they finished cleaning them, so that should a guest arrive, the receptionist would be

able to check them in, or inform them the room is ready without having to wait for housekeeping to report to reception once all rooms are clean.

- **Guest self check In** :
  When I was designing this application, I made it possible for a new user to create an account. In doing this they are automatically set as a guest until the manager would allocate them more responsibilities on the system. So in doing this, if the guest was to arrive and there wasn't a receptionist at the desk at the time, they could simply log in, and check themselves into the hotel. This would make check in more efficient, but in budget hotels remove the need for a receptionist at all. making the running costs of the hotel much cheaper.

- **Channel management** :
  A final feature I would really like to integrate into the project should I develop it further would be to integrate a channel manager into the system so that should a booking come in through any of the online booking agents, such as Booking.com or Expedia, that the reservations would seamlessly upload into the system, and reduce the availability and have all the clients details in the database. This will make the reservation teams job a lot easier, and remove the possibility of human error.

So in conclusion, I am very satisfied with the outcome of my project. I feel as though I have created a project that covers the specifications that I had wanted. I really feel that I have benefited myself through working on this project, as I have learned a new language for the creation of this project, but have also worked with firebase, which is now such a popular database and hosting system to use. Although I had great experience from my four years in college in Galway Mayo Institute of Technology with the various languages and databases we have used, I felt as though I should expand my knowledge and prove that I am capable of applying the skills that I have learned and create this project in completely new ways. This project has also benefited me massively in my own working career, as my manager has gotten more of an interest in how to run the reception and reservations in a more positive manor, and a better understanding of the true possibilities of running Micros Opera to its full potential.
I would like to take this opportunity to thank all of the staff at Galway Mayo Institute of Technology for the great support that they have provided over the past four years of my studies, and in particular Mark Campbell for all of his help over the past year. I would also like to thank Lough Rynn Castle, Ballynahinch Castle, Clifden Station House, Renvyle House and Forster Court Hotel for graciously allowing me into their beautiful hotels to investigate their various systems, and meet with their brilliant staff. In the future I do plan to continue developing this application to include the opportunities that I outlined above, in the hopes of showing it to my manager, or other hotel owners and receive their opinions, with the possibility of selling on the application for professional use, and also use this project for potential interviews in the future to display my knowledge and capabilities to employers.

Bibliography

[1] Muhammad Tahir. Performance assessment of agile methods. *technology*, 1:11–21, 2019.

[2] Wikipedia. Agile definition and overview. *informative*, pages 1–2, 2020.

[3] Hoist Group. Hotsoft hotel software. *Software*, 2020.

[4] Okpe Kelechi Thecla Scholastica Chukwu Nwaohiri Ngozi, Maria and Ifeyinwa Okoye. An analysis on the impact of the usage of fidelio opera property management system in transcorp hilton hotel abuja. *software*, 6(2):1–19, 2018.

[5] RoomMaster. Roommaster hotel software. *Software*, 2020.

[6] Agilysys. Agilysys visual one hotel software. *Software*, 2020.

[7] Guido Albertengo. On the performance of web services, google cloud messaging and firebase cloud messaging. *technology*, 6:31–37, 2020.

[8] Vipul Kaushik. React native application development. *technology*, 4:1–21, 2019.

[9] H. M. Moyeenudin. Data management with pms in hotel industry. *business*, 1:1–2, 2018.

[10] M.J Van Den Haak. Exploring two methods of usability testing: concurrent versus retrospective think-aloud protocols. *technology*, 1, 2003.