Section 9 - Evaluation

Table of contents:

Computing Languages

Languages Chosen

Python

The main reason why python was chosen as my programming language was due to my experience using the language. An advantage of using this language due to my experience is my knowledge of the extensive libraries python has to offer. Not only does python have many libraries that are pre-downloaded but it has also allowed me to download other libraries. These libraries assisted me in every aspect of my project from the GUI to sending emails. Another advantage of Python is the ability to process data with ease compared to other programming languages. The ability to use OOP (Object Orientated Programming) has also been a very helpful feature as this has allowed me to create user view and access levels with minimal code repetition. Another feature of python that has allowed me to reduce my code repetition is the ability to create user defined subroutines, without this feature my project may have not been possible at all.

One disadvantage of python is the limitation of creating a user interface. Python is very limited in its capacity to create aesthetically pleasing user interfaces and therefore I had to use external software such as ‘Page’ (discussed below under Tcl) and therefore I struggled with this aspect of the design.

SQL

This language is used to create and edit the database. This is used extensively throughout the project due to the nature of connecting to the database. An advantage of using SQL is that it is very close to the English language and therefore does not require much syntax to learn. Another advantage which was crucial to the success of the project was the ability to use this language within Python through the library sqlite3.

A disadvantage of using SQL is that, when using sqlite3, each record that is returned is stored as a tuple within a list. This becomes a problem one only one field is being returned and requires extra processing to ‘clean’ the data so it can be processed by the system. Another disadvantage with sqlite3 is that all SQL commands are required to be entered in as a string and are not flexible and therefore functions needed to be created to effectively ‘edit’ sql commands to make require less code reusability.

Tcl

To create many of my interfaces I decided to use a piece of software known as ‘Page’. This software is a ‘drag-and-drop’ GUI creation software and allowed me to style each widget and to place this widget exactly where I wanted it to be placed. This software was very useful when creating the login screen specifically placing the ‘here’ button right next to the text and using the same font as the text. The programming language behind ‘Page’ is known as Tcl and when a ‘Page’ template is saved it is saved with this extension. To use ‘Page’ within my project I had to generate python code from this template which is a built in function within this software.

One disadvantage of this software is simply the code that it generates is complex and difficult to edit. As ‘Page’ does not have certain features such as drop down menus this had to be edited manually which was made difficult with the complex ‘Page’ code. Another disadvantage is that none of the widgets are self-declaring and therefore needed to be changed to assist individuals maintaining the code.

Html

To style the emails I have decided to use html as this allows me to have more control over the design of the email rather than using plain text. Due to the style of the email which I wished to create I have used very limited tabs and this style could have been achieved via plain text. The reason which I decided to use html for this project was to assist the maintenance staff in the future if at some point the design of the email needed to be changed for an increase in user friendliness. This would have not been possible if plain text was used. A disadvantage of using this is the added complexity in which using html has caused.

Purpose/objectives delivered

As mentioned above to achieve many of my objectives I needed access to Python’s many different libraries to complete many objectives. One library that was particularly useful was the ‘pickle’ library as this helped me to implement my periodic backup without the disadvantage of the user being able to change the month that was previously backed-up through a text document. Other libraries that assisted my programming include ‘matplotlib’ allowing me to create graphs, ‘docx’ allowing me to create a word document and many more libraries.

Effectiveness

I believe my choice of programming language has been very affected through the use of many of its features including the ability to use ‘Page’ and the different libraries that Python has to offer. Using ‘Page’ I believe was very effective as this allowed me to quickly create user interfaces that are more user friendly than if I did not use ‘Page’. This is taking into account the added complexity that using the code generated from ‘Page’ has added to my system. Although my use of html within my system is not very effective due to the fact that there is scope for the design of the email to be improved using this html I believe that this was worth undertaking.

Comparison to Other Systems

Sirvoy

Similarities

My system and Sirvoy have many similarities within functionality but the scope of this functionality is much smaller within my system. One piece of functionality that both systems have automatic emails sent out to the user and that both systems allow the creation of different ‘room types’. Another similarity which my system has with Sirvoy is that both have different access levels depending on the user that has logged onto the system. The data stored within these systems are also encrypted.

These similarities to an existing system has demonstrated that my system has key functionality that is useful and that makes it competitive in the hotel management system market.

Differences

Although my system has many similarities with Sirvoy, as listed above, it also has many differences. One of these differences is that my system is able to create a ‘tape chart’ which allows the hotel to easily check which rooms are booked at certain times. Another difference is that my system does not restrict the user in how far in advance bookings can be made whereas Sirvoy restricts this to a year and a half in advance. This perhaps would have been more beneficial for the hotel staff that is using the system as staff would not need to be distracted by bookings 3 years in advance that is ‘blocking’ other bookings within the system. This also acts as a form of validation as the customer may not intend to book their holiday several years in advance. One advantage Sirvoy has over my system is that Sirvoy allows the hotel staff to customise the different fields that they wish their customer to enter.

These differences within the system demonstrate that my system still has many improvements that could be made to it and how my system contains some functionality that some professional systems have not implemented.

Weaknesses/Limitations

Objectives

All objectives have been met apart from the objectives which I decided to not implement discussed in ‘Section 5 - Post Prototype’ and the objectives in which I decided to change (instead of sending an email when a record is requested, just displaying this in the system was better) are discussed in ‘Section 5’ as well.

Discussion of weaknesses

One main weakness which I have mentioned throughout this project is the user friendliness of the system. This has been the aspect of the project which I have struggled with and one in which I predicted that I would struggle with during the discussion stage. Even with the assistance of ‘Page’ and the advice from my peers it is clear that the main weakness of my system is the user interface design. This impacts the user as they may not be entirely clear on how to use certain features of my system and may struggle to read the graphs within my system.

A very clear limitation of my project is the objective points which I have not implemented. Although these were not crucial as discussed in ‘Section 5 - Post Prototype’ these would have certainly made my system much better for the user as this would have had more features.

Another weakness of my system which I would like to mention is that of the ‘reminder email’. When a reminder email needs to be sent this is done before the user interface is generated and therefore the user needs to wait for this to be done. The implications to the user is clear as they will need to wait for the login screen to load. If the email is not able to be sent for any reason the user will need to wait for a ‘TimeoutError’ to occur which takes a considerable amount of time.

My encryption is limited in scope and it is not very strong. A very large weakness related to my encryption is how I implemented editing records. In the way that I have done this when editing a record each field corresponds with an entry widget and therefore to prevent the user form needing to re-enter their credit card details I have decrypted this data (see perfective maintenance for potential solutions). This results in the disadvantage that when editing a record the encryption has been bypassed. Yet another disadvantage of my encryption is that due to the technique being used (caesar cypher) it is relatively easy to break the encryption if the key is discovered or by looking at common symbols that appear i.e. the most common letter in the English alphabet is ‘e’ and therefore the symbol that appears the most represents ‘e’.

Potential Improvements

Corrective Maintenance

There are several errors throughout the system which need to be fixed. One of these errors is whenever a coma (‘,’) is entered into the system when searching, editing and adding records. The error is caused by the fact that SQL commands use a comma to separate data being entered. This is not a large issue as commas are not required in any field but may cause confusion when entering the ‘salary’ field as many users may enter a comma due to the large number for example ‘,’.

Perfective Maintenance

When a customer is editing their bookings they are able to edit everything including the price of the holiday. If the customer changes the number of nights they stay at the hotel this will not update the price field. This issue occurs when both the receptionist and manager edit and add a record to the ‘Booking’ table. To aid user friendliness and prevent confusion the same processing of the data should have been implemented to these users as what was given when a customer originally books a holiday as this would recalculate the price of the holiday. Sadly due to time constraints I was not able to do this but, as the code already exists within the system, this should not be overly difficult.

One aspect of my system that could be improved is the issue with the reminder email as discussed in the weaknesses section. The issue where the user needs to wait for reminder emails to be sent could be solved with the use of threading. I experimented with threading in an attempt to change the cursor when the program is loading but when this did not work I did more research and discovered that tkinter (the library used to create the user interface) does not support threading and therefore this was not possible. Due to time constraints, this prevented me from attempting to use threading to fix this issue although this may not work just as changing the cursor was not possible. The library used for threading in Python is simply called ‘threading’.

When a staff member is editing their own record they are able to edit the ‘salary’ field which they should not be able to do as they may increase their own salary. This could be fixed by adding an if statement when creating the interface checking the class that is currently being used and removing this field. I was not able to implement this extra piece of functionality due to time constraints.

An improvement to the system if images of rooms could have been included instead of the user having to ‘guess’ what the rooms actually look like. The ‘Facilities’ table and the ‘Room_Type’ table could be included when a customer is booking a record to aid user friendliness.

A change I would make to my system is to replace the ‘Quit’ button, once logged in, to a logout button. Throughout the project I have gone back and forth on whether to include both buttons or which one. The reason why I did not include both is because I did not have room to do this within the ‘manager main screen’. I made the decision to keep the ‘Quit’ button as only one user should have access to one account and therefore should not need to log out and due to time constraints. Upon reflection a user may log out of the system to ‘lock’ their account or to let another user login using the same computer. Although the user could simply quit the system and restart it this is not user friendly.

Adaptive Maintenance

Throughout my research I discovered that many hotel management systems provided functionality that would link the system with external websites such as booking.com and expedia through using an API. Both my technical knowledge and time limitations prevented me from implementing this piece of functionality.

Review of Project

My Own Strengths and Weaknesses

Strengths

One strength of the project was my ability to edit code that I have written in the past and reuse this for my project. Being able to do this taught me the importance of code reusability as it may save time on future projects just as this has done with mine.

Another strength of the project was the research which I conducted during development as this allowed me to implement many features which I would not have had the knowledge to do otherwise. This can be seen by the extensive list of references within ‘Section 6 - Software Development’.

The decision to use a semi ‘Agile’ approach to this project I believe was beneficial as this allowed more functionality to be completed than what would have been completed if a ‘Waterfall’ approach was taken. Due to the flexible nature of the ‘Agile’ approach I was able to very quickly and easily change or remove functionality depending on time constraints. This was very helpful after the feedback that was given from ‘Section 5 - Post Prototype Refinement of Design’ as I did not need to change much code and the code I did need to change was changed quickly. The flexible nature of this approach also assisted me in the previous section ‘Section 4 - Prototype’ as I was able to select different objectives and use this code within my final system.

Weaknesses

One weakness within my project is my artistic ability through creating a ‘good’ user interface. As mentioned above and in previous sections and can be seen even in the design of these documents one of my weaknesses is design. Through feedback from my peers I have attempted to minimise this issue.

Another weakness of my project is my time management skills through this project. This is further discussed in the Impact of COVID-19 section and shows my unpredinness for the movement of the deadline. Disregarding this, there was still features and ‘finishing touches’ to both the documentation and the system which I intended to do but would not have had time to complete whether or not the deadline was moved forward or did not move.

I have discussed the benefits of the ‘Agile’ approach in the above section but the main drawback of this approach is the lack of documentation throughout the project. The difficulty of this was not having completed sections of the documentation until the end of the project which caused issues with time management.

Changes I Would Make

One change I would have made after completing the project would be to focus more on the documentation of the project. Specifically I would have focused more on the ‘Section 2 - Investigation’ documentation as I could have included more detail in the systems which I researched and I could have researched even more systems. Another section which I would have spent more time on is this section ‘Section 9 - Evaluation’ as I have much more to say specifically in the Potential Improvements section as I had many ‘finishing touches’ and extra functionality which I wish I could have implemented in which I could have talked about at length. Sadly due to the deadline being pushed forward mentioned in the section below ‘Impact of COVID-19’ I have not had time to discuss. I would have also spent this extra time and focus on making these documents look more professional and easier to navigate.

Another change I would make is to comment my code while I was creating it instead of retroactively. I did do this within certain parts of my code and it has proven very helpful to have well commented code when going back to code which I have not looked at for extended periods of time.

As I have mentioned several times within this document I have struggled with the time limitations and therefore I would have created a ‘planner’ of some description to dedicate more time to this project and to specific sections.

Impact of COVID-19

In light of recent events surrounding the current outbreak of the virus of ‘COVID-19’ I would like to discuss the impact this has had on this project.

The impact this has had on this project was the pushing the forward of the deadline of this project. Although this may show poor time management skills on my part which if I was to do this project again I would certainly improve.

The two main aspects of the project which were affected was ‘Section 6 - Software Development’ and this section. The section which has been affected most by this has been this section and is why this document looks incomplete in certain sections. Within this section due to the deadline being moved forward I had to not include the full list of my objectives and which objectives were fully met, partially met and not met. Instead this is replaced with a summary addressing that these objectives have been discussed in previous sections. The ‘Comparison to Other Systems’ has also been affected as I was planning to include more comparisons but due to the new deadline I was not able to do this and ‘Potentials Improvements’ has also had to been shortened as discussed in the ‘Changes I Would Make’ section.

The deadline being pushed forward has affected all aspects of the coursework but in all other sections only ‘finishing touches’, as making the documents themselves more aesthetically appealing and adding more detail in parts, were affected. To be clear the creation of the system was not impacted by the movement of the deadline but corresponding documentation was including the commenting of this code although all the code should still be self-documenting (names of classes, variables and functions) as this was not affected.

I decided to include this section to discuss what has changed with this project due to the spread of this virus. I believe this event has allowed me to experience a deadline being pushed forward, which is an event which occurs frequently within the IT sector and is an event I should have been prepared for.