GCE COMPUTER SCIENCE H446A COMPONENT 03

Harry Alderton

Contents

[Analysis 2](#_Toc477792057)

[A description of the problem 2](#_Toc477792058)

[Identify all the stakeholder 2](#_Toc477792059)

[Justify why the problem can be solved by computational methods 3](#_Toc477792060)

[Research 4](#_Toc477792061)

[Features of the proposed solution 10](#_Toc477792062)

[Software requirements 12](#_Toc477792063)

[Hardware requirements 13](#_Toc477792064)

[Success criteria 13](#_Toc477792065)

[Limitations of the proposed solution 14](#_Toc477792066)

[Design 15](#_Toc477792067)

[Decompose the problem 15](#_Toc477792068)

[Structure of the solution 17](#_Toc477792069)

[Algorithms 21](#_Toc477792070)

[Usability features 22](#_Toc477792071)

[Key variables and structures 23](#_Toc477792072)

[Test data for development 26](#_Toc477792073)

[Test data for beta testing 28](#_Toc477792074)

[Iterative developing of the coded solution 29](#_Toc477792075)

[Iterative development 29](#_Toc477792076)

[Errors in development 55](#_Toc477792077)

[Prototyping 58](#_Toc477792078)

[Annotated modular code 59](#_Toc477792079)

[Testing to inform development 70](#_Toc477792080)

[Reviews 78](#_Toc477792081)

[Evaluation of solution 79](#_Toc477792082)

[Testing 79](#_Toc477792083)

[Usability features 79](#_Toc477792084)

[Evaluation 79](#_Toc477792085)

[Maintenance 79](#_Toc477792086)

[Bibliography 79](#_Toc477792087)

# Analysis

## A description of the problem

The problem that I’m going to solve is to help the client’s independent business in which the client runs alongside parenting and a job. The business is a housing renting; this involves customers having to look in the local newspaper ‘The Huddersfield Examiner’ and specifically look in the properties to rent section or having to physically drive past the properties to see the to rent/let sign. Using this technique to find renters is very impractical for both those wanting to rent a property and the client trying to rent out the property. This is mainly due to newspapers moving online therefore wouldn’t be as easy to find the property to rent and sell section. Further putting ads in the paper costs money and posting frequently it becomes quite expensive when there is a much cheaper alternative of a website which would be seen as computational methods of solving this particular solution.

Another problem which the client faces is having constant phone calls from renters or potential buyers trying to book the viewings, which requires a 5 to 10 minute phone call to book a time slot and select the correct date which along with parenting and a job causes problems which takes up more time than it should due to this being inefficient and a very old fashioned way of going about bookings. Even after booking times etc they also then have to have a separate document which is usually in excel which involves the client manually having to type in the details of each person which has an interest which takes up time along with all the information being stored in the same place as this could be at risk of the computer which the software is runs on from crashing which you could lose all the vital information which wouldn’t look professional also manually typing all the details out isn’t very time efficient.

The problem continuous when there’s interest in a property the potential renter has the make contact with the clients and the client uses a diary of the dates of meeting, this in the past has caused problems as having all the dates and times all in one place losing or the data being stolen causes massive problems as the client is unable to recall when the viewings are and therefore has to call the potential renter and explain which is not only time consuming and inefficient it also looks unprofessional and could potentially put the renter off.

Currently each renter has to keep track of when their payments have to be paid, which with each contract being different and each month changing renters can be late with payments and therefore face penalties and then put the client behind on payments to the bank etc. which overall causes inconveniences for both parties involved.

There is also a problem with there only being one picture of the property, which can be rented in the newspaper, and if the potential renter drove past the property then they would only be able to see a glance, which means that my client gets a lot of people who view the property that don’t actually end up renting due to they haven’t seen the inside and only had a very limited description of the property which this then becomes a waste of time for both parties involved.

## Identify all the stakeholder

A stakeholder is a person, group or business who is concerned or interested in the idea or concept of the website idea. Those concerned maybe:

**Users of the website** – They will be concerned for the website to find properties which are to rent. They will also be creating a user to log on so they are able to book appointments, leave feedback, ask questions and rent the properties. They will want it to be a user friendly website which is easy to navigate and use. They would be interested due to how the current system works which is paper based and phone calls which uses their time and incredibly inefficient and makes them think that the business isn’t going to be reliable when they get phone calls due to loss of information on my client’s behalf.

**Hosting Company** – They will be running the website from their own servers along with providing the website with a domain name. Also if the website crashed due to server errors for example server got too hot and shut down or they server got hacked they would be responsible to fix the problem this gives them the interest in the website as they want the website to be fully working as they will keep my client which increases their revenue from a repeat customer.

**Client** – is a stakeholder in the website as they have direct access to the website and they will be using with the means of showing the properties to potential renters and therefore would want the design be user friendly also with the way which the different features are used to help improve his efficiency in running the business, due to currently the system the client uses paper based recording system along with storing all the information in one place which has the risk of losing all of the information. This is very inefficient and wastes time especially when he has to ring back due to the loss of data.

**Administrator** – This could either be the client or the creator of the website depending. More than likely it would be the creator of the website due to them knowing the code behind the website and it would be more convenient, they are a stakeholder as they have an interest of keeping the page up to date with information along with keeping the client happy.

**Programmer** – They are responsible for meeting all requirements that the client has set. With the code behind the website having to be operational with no major glitches therefore is a stakeholder as they want to complete the task to the best of their ability’s keeping the client happy.

## Justify why the problem can be solved by computational methods

The property website is needed mainly due to the impracticality of the current system which my client currently uses which involves as explained in the ‘description of the problem’ section, the client uses word of mouth along with newspaper adverts which is a dying industry with less and less bought as its all moved online however seeing adverts for properties to rent are very infrequent and are more than likely buried in the website which makes it extremely difficult to find unless you particularly search for the section, therefore as a solution the problem can use computational methods mainly from using a website which has all of the clients properties on which makes it much more efficient for potential renters to see the properties much more clearly. This solution also is more cost effective as posting adverts in the newspaper is expensive and although it would cost money to host the website in the long run it would be much cheaper as you can post properties on the website for free without having to pay a fee.

The problem of having all the dates of meetings stored all in one place in a diary which if stolen or lost then all then the client has to rang back, however this can all be fixed using computational methods as you would use an online dairy on the website so the renter is able to log in and see when the client says he is prepared to do property tours, the potential renter is then able to select the time slot on the calendar which is then automatically updates and when the client logs in they can see which property is wanted to be viewed at which time slot therefore cutting out any risk of using a paper based system. This not only makes it easier for renters it also makes it much easier to track the amount of viewings for the client.

A computational method is also needed as the client receives multiple phone calls with enquiry’s which take up time which isn’t very efficient when the enquiry can be done within the website through either an enquiry page which can then be emailed straight to the client who can then respond in their free time, there could also be a FAQ page which might answer an information which the potential renter might have which will then stop any frequently asked questions which occurs which the client currently has the respond to. The way which the potential renters books using computational methods is by using a form which is created while creating a new user which involves their name, address and contact number so when they select the correct time for them the information is inputted into that slot which then the client will be able to see all of their relevant details rather than using the current system which involves the client having to type out all of the data for each potential renter and store it on his local system which has to potential to be corrupt or accidently deleted, which a website wouldn’t have this problem as its stored on secure servers with backups just in case anything did happen.

The way which payments can be made easier for the client and the renters is using computational methods is though using the calendar which can only been accessed once logged in, this calendar simplifies all the payments for each property into one calendar which will display when payments are due. This creates a much easier system rather than using phone calls or texts, as the renter is able to see in advance and work out their finances through this.

The final way, which the problem can be solved through computational methods, is the photos of the properties as currently there is only one photo placed in the advert section of the newspaper and even if the potential renter walks or drives past the property they only see a quick snapshot as they cannot see any of the rooms, which though the website there would be more photos of both inside and outside of the property with a floor layout plan, this gives the potential renter a fuller picture. This prevents anyone who would waste time and isn’t fully interested therefore making the business more efficient and time effective.

## Research

The first part of my research was an interview with the client. Before the interview commenced I explained about the project to design a website and explained the kind of questions I would be asking so then the client wouldn’t be overwhelmed with how to answer and give them some time to think about what they wanted. I further explained that this was going to be one of a couple of interviews which was going to happen, with this first one being just an overview of the website they want creating along with some key information so I would have a clear idea of how to design the website.

The first interview:

Me: Hello sir thank you for having time to have a short interview with me, it shouldn’t last too much longer than 10 minutes. From this interview I will be asking questions which will help guide me towards making a website fit for your needs.

Me: First of all how do you advertise your properties to rent?

Client: I currently use the local newspaper ‘The Huddersfield Examiner’ or have a sign up outside of the property. I know this is very old fashioned way of advertising I am also aware that not many people stop and actually look when driving past the signs.

Me: When a potential renter has interest in a property how do they contact you and how do you record details?

Client: Currently potential renters get my mobile phone number from the advert and they call me, which usually involves a 5 to 10 minute talk. I record their details on a piece of paper which I transfer onto excel.

Me: Has using this method caused any problems like lost data or system crashes?

Client: Currently the problem I have faced has been typing all of the data onto excel however I have recently bought a Mac which doesn’t support excel and I don’t have Microsoft Office installed so I can potentially see a major issues happening when I try transfer this file onto my Mac.

Me: How do you book viewings for the properties?

Client: When a potential renter rings me and requests a viewing I give them the date I am free and give them a couple of times frames to choose from and let them decide. Once they have decided I fill in my dairy the time and whom I’m meeting.

Me: Have you ever faced problems with way of managing this form off bookings?

Client: I have in fact had many problems with this, I have lost my diary many of times which has caused problems with remembering what days I have bookings and what times, which then I have had to call back to ask the potential renter that time and date we set for the viewing which has then made me look unprofessional and also wasted a lot of my time.

Me: How are payments currently managed?

Client: The renter signs an agreement, which says payments, must be paid every 28 days, which works out to be every 4 weeks from when they moved in. The way I keep track is using my diary.

Me: Have you found problems with payments being made on time?

Client: Yes every month I find one or two renters who are late with payments which leads to late fines that I hate giving out but it’s due to the bank charging me.

Me: As explained before you said you advertise in local newspaper, do you only display one photo? If so have you thought this has led to time wasters or even putting people off?

Client: Yes currently I only display one photo, which properly does put people off, as it doesn’t give a full picture of the property.

Me: Now can I ask some questions about your business? This will be used to design your website.

Client: Yes sure go for it

Me: What’s your business called?

Client: Daisy Lee Holdings

Me: What do you want the website to do?

Client: I want all of the issues which I explained earlier to be solved and if possible for them all to be digitalised and put on the website, so I would like all of the properties to be displayed on the website with multiple images, floor layouts the address and so on this would prevents anyone wasting my time and they have a clearer idea of what the property is. If possible I would like a calendar which has all of the important dates on for each flat which can only be viewed when logged on so not everyone who views the website is able to see this information. Further I would like this calendar to have appointments for viewings which once logged on they can select the free time slot, which they are able to select and book onto.

Me: Have you thought about a general enquiry’s page along with a FAQ section?

Client: Yes I was just thinking about that and think it would be a very good idea because it would save me a lot of time to answer any questions they have using emails as I can do this when I’m free. Also the FAQ page would be very useful because then any questions I notice are frequent I can place here and this would be more efficient.

Me: I’m now going to ask what you want your website to look like. What would you like your heading to be?

Client: Daisy Lee Holdings

Me: What would you like your navigation bar to be, there’s a drop down navigation bars or you could have a side navigation bar?

Client: I would like a drop down navigation bar if possible please.

Me: What would you want the navigation bar headings to be?

Client: Property’s, Calendar, a sign in/ new customer sign up, Contact page, a FAQ page I also thought maybe a testimonials section of feedback from past tenants.

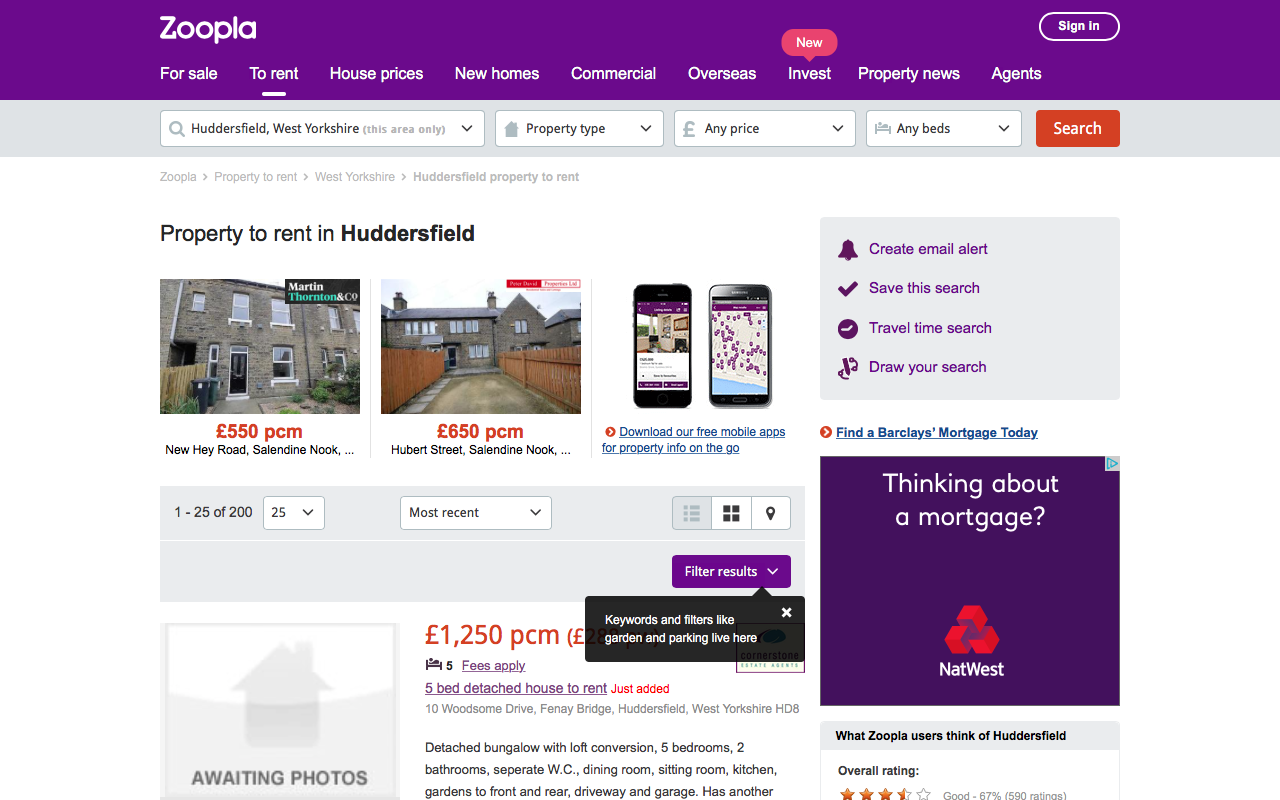
Me: No problem I will do that. My final question for you is what style of a website would you like, maybe a classy or a friendly style?

Client: I would like a classy colour scheme, which catches the eye; I will leave the colour scheme up to you.

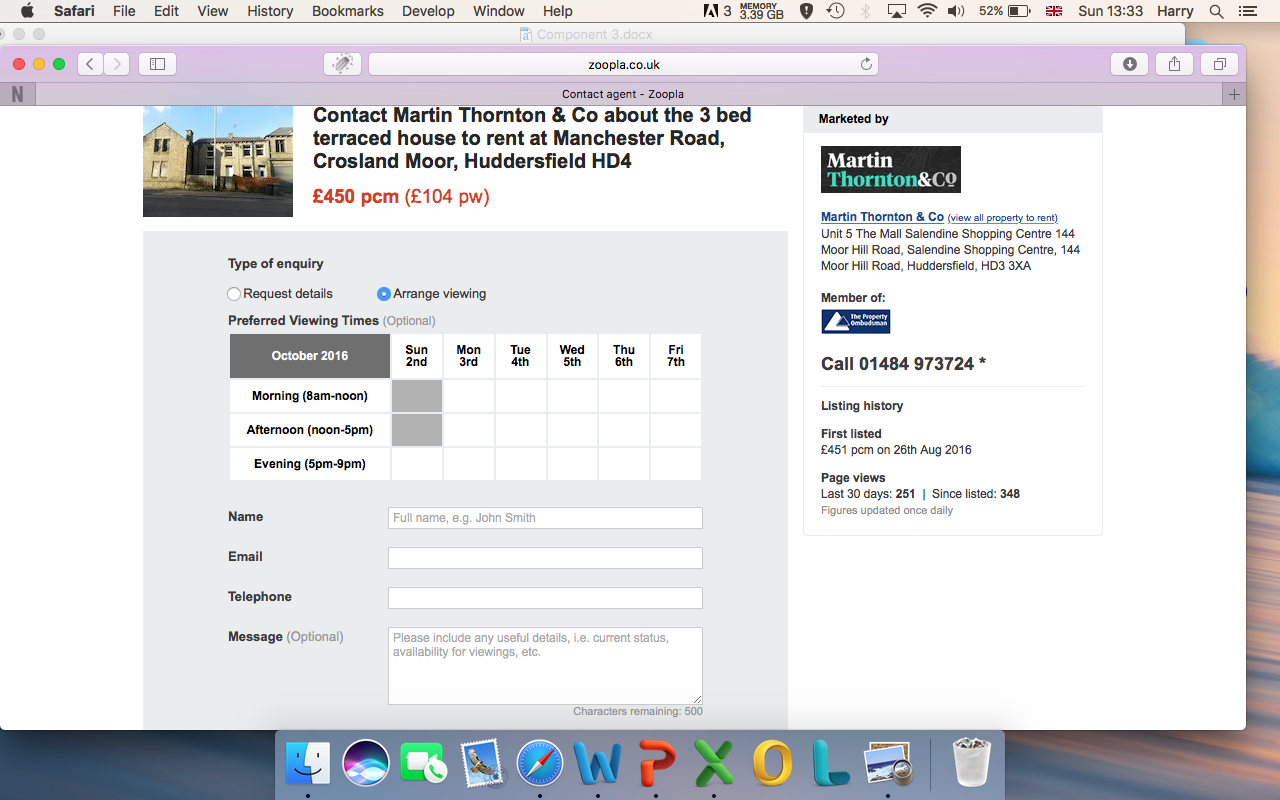
Me: Thank you for meeting me, I will be in touch in the next few weeks that will allow me to create a mock up design for you to view and recommend any changes so then it fits your needs and wants. Thank you once again.

Other Property Websites:

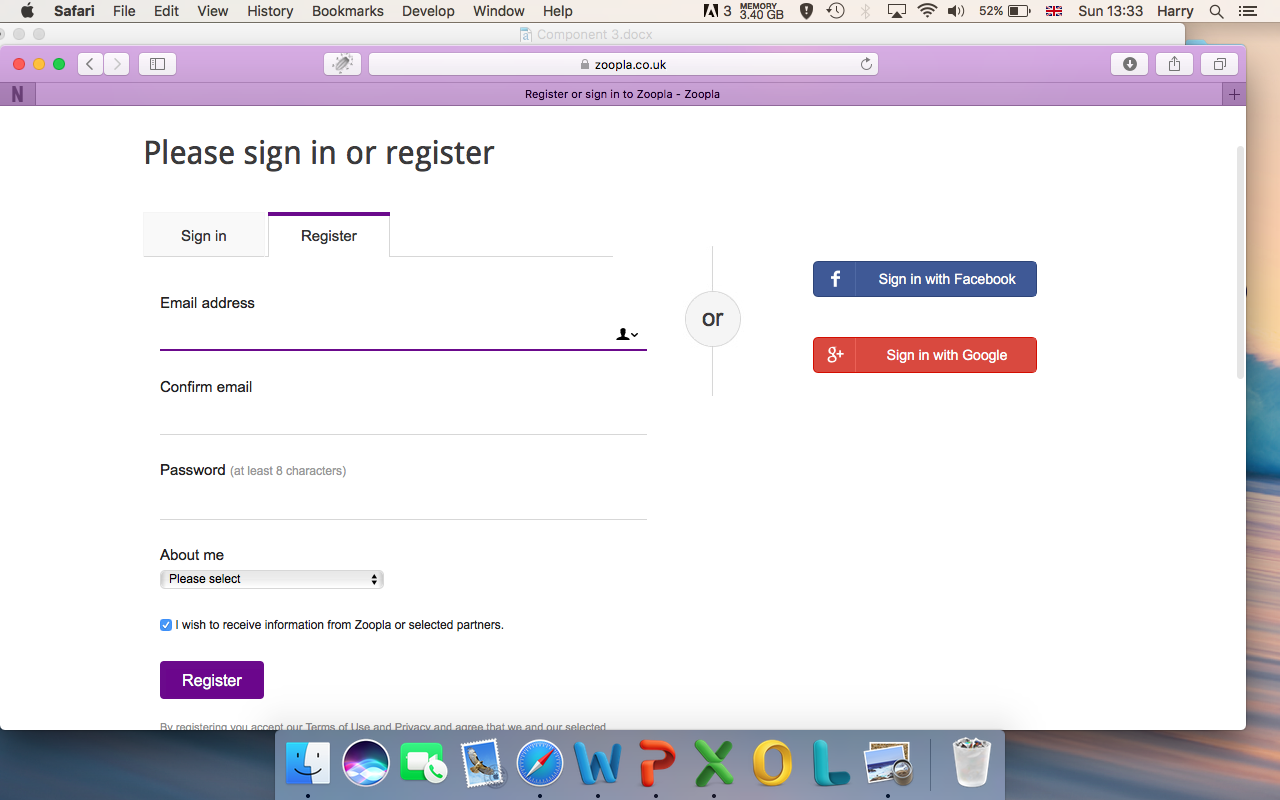
For my research I decided to look into another website, which is a housing renting websites. However the company is very commercial and offer other services which my client doesn’t need therefore I am just going to look into the properties to rent sections along with the layout and the login system to see the style and the kind of information they look for. (Zoopla, n.d.)



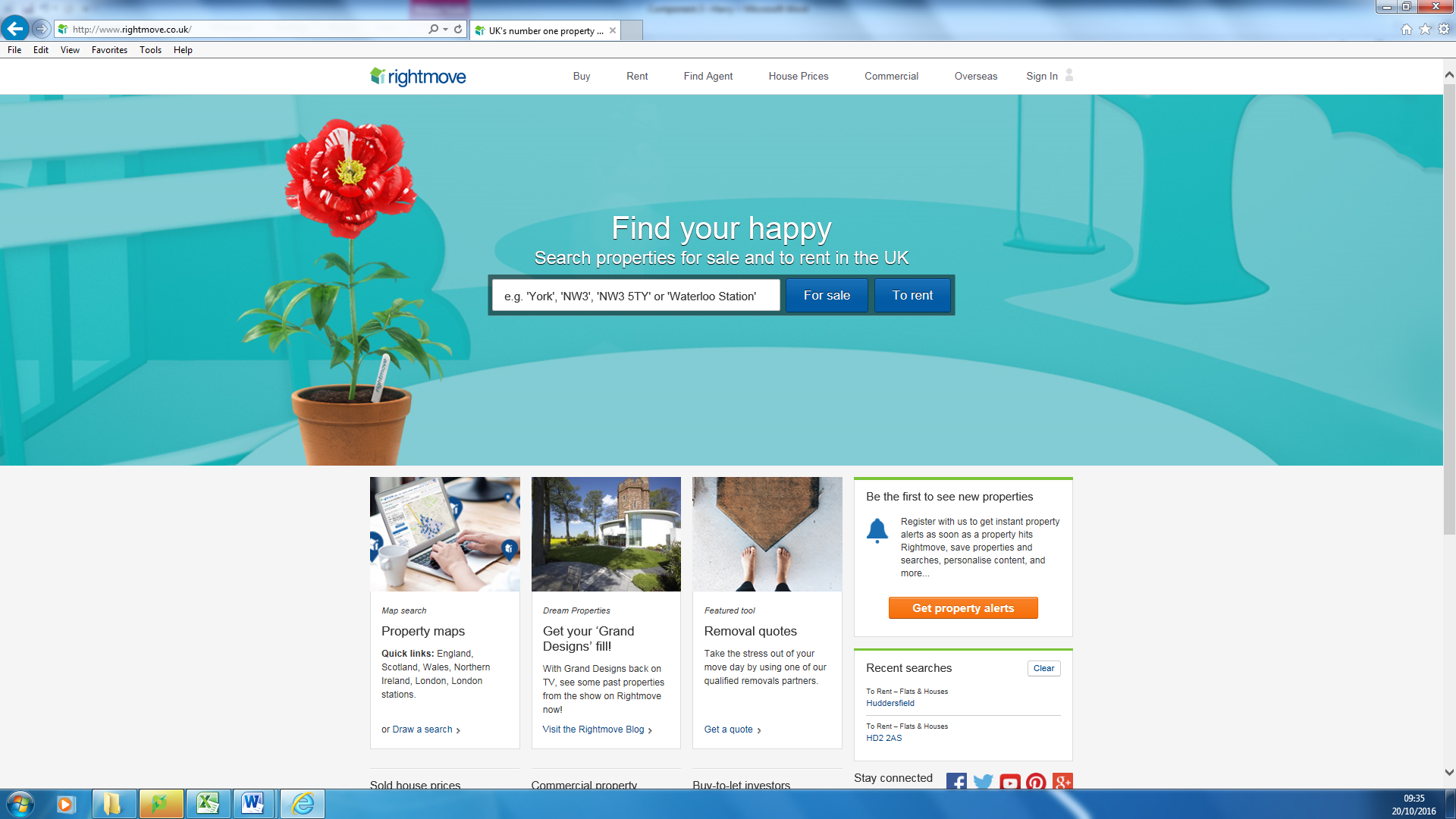
Zoopla is a very popular commercial website which advertises houses to rent and buy all over the country. The layout for Zoopla is very simple and easy to use; this makes it easy enough for any person whether they have tried to use renting websites before. Their system also includes parameter which allows you to search; however for the clients website there will only be a few properties on at a time therefore wouldn’t need a search parameter.

I also like how the key information like the price is in a bold red colour making it stand out and easy on the eye, this also allows quick searching which could increase customer satisfaction on the website.

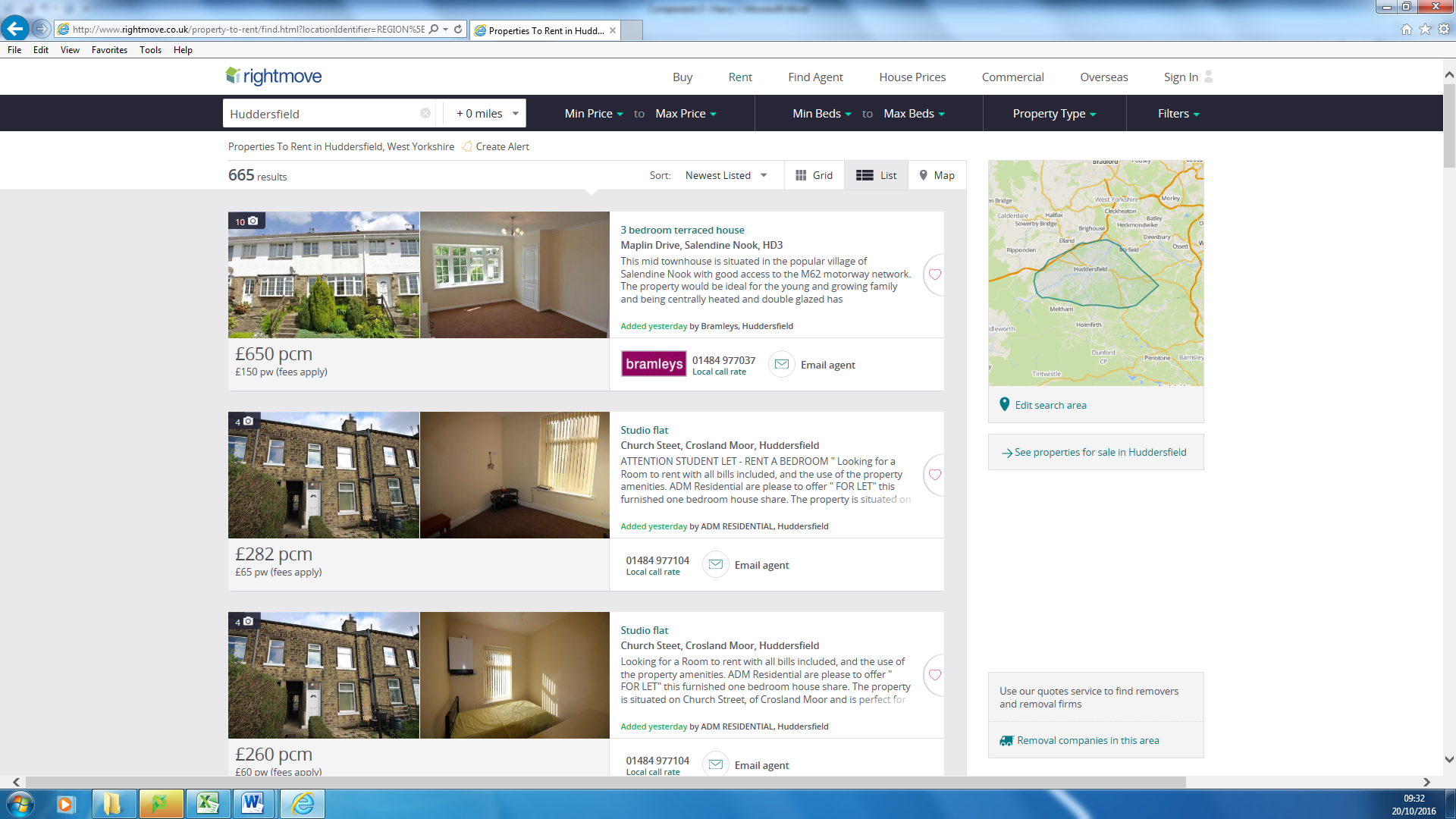
This section is another feature which the client wants putting into the website, which is the booking system to view the property. This section allows the potential renter to view the property at the time that suits the owner of the property; this is very simple and effective which allows them to fill in details that will alert the renter. My client realistically wants something like this but only able to choose a time once they have created an account. This style of booking is the best which I have seen while looking on other renting sites like Right Move which doesn’t have this calendar set up which is realistically what I want my website be styled like.



Zoopla also allows people to register using a simplified method by just using email and a password. For the website I am going to use something along these lines however their name will also be used to create an account. However to log in they will use the email and password. I like how Zoopla have made it really simple for the input of each component and it is aesthetically pleasing.



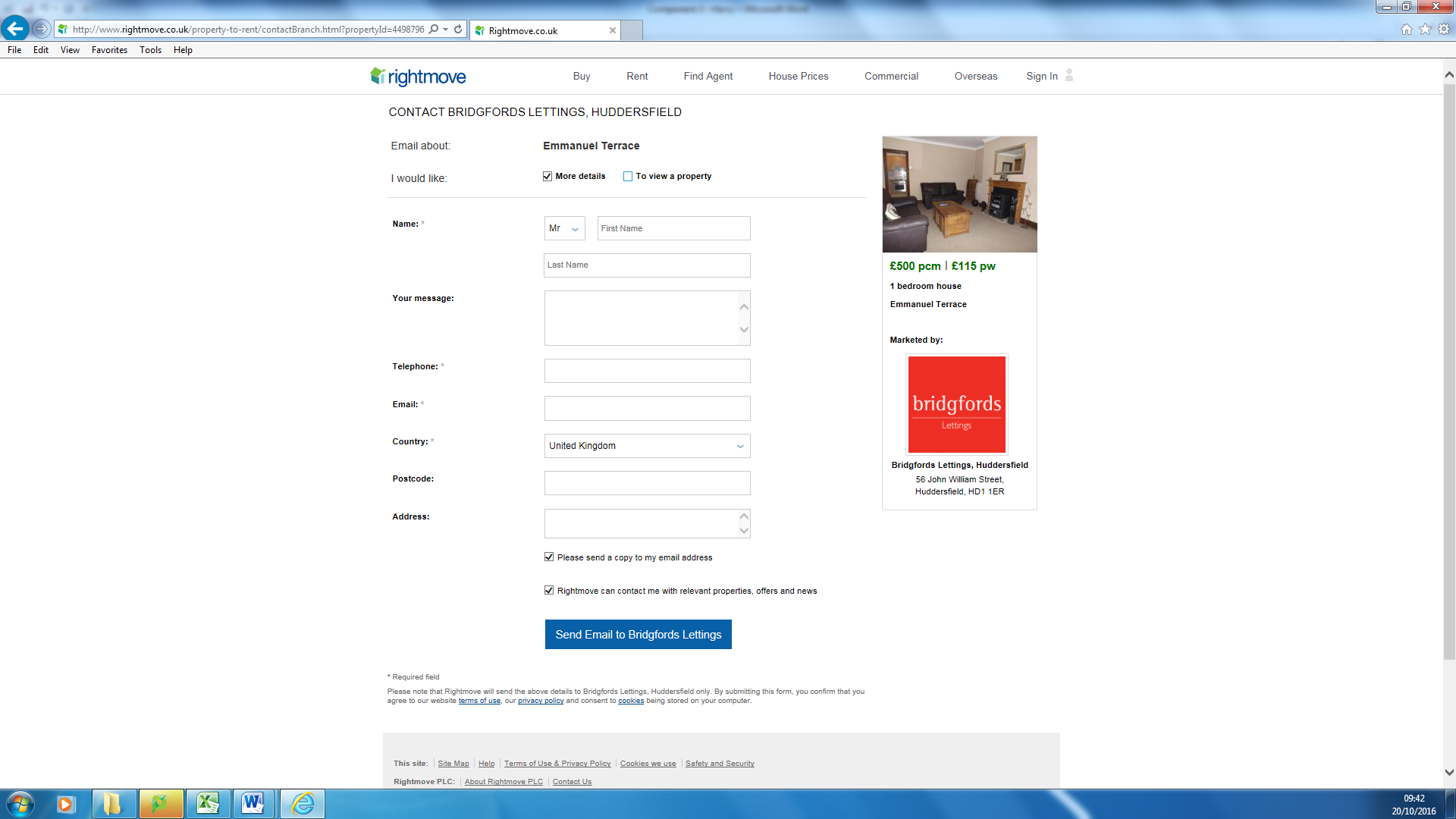
This website is Rightmove which says it is “The UKs largest property portal” Which is similar to Zoopla which allows you to log in, create users and most importantly requires you to look at properties.

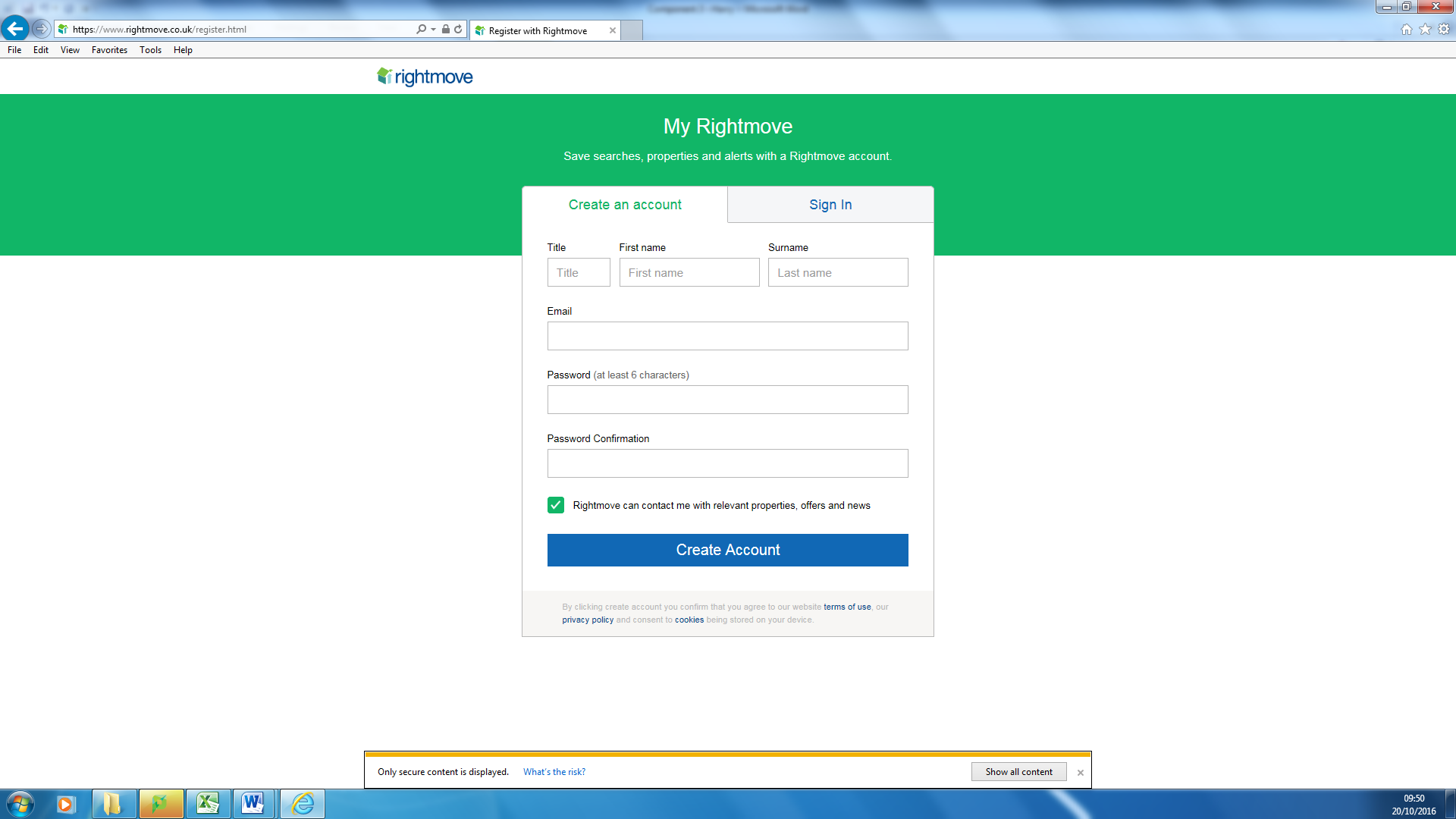


Right move allows users to search using parameters for properties based on price and location, however the client won’t need this due there only being a few properties on the website at a time.

The style of Rightmove is very simplistic and very easy to use with a simple colour scheme which doesn’t draw the eye away from the properties. The design is very user friendly with bold text and large buttons which makes it easy to click on features.

I also like how the properties are displayed on this search section with a snapshot of pictures of properties along with some description and address, this kind of style is what I was thinking of for my client’s website.

This is the enquiry form for the properties which is very basic but has all the details that will be needed. Also rather than having a calendar system for viewings which I was thinking of including and what Zoopla were thinking they just have a tick box if they want to be contacted back about viewings, this could work for Rightmove however for my client he doesn’t have time for contacting back for something which can be filled on the website.  
For the login system it uses a similar style to Zoopla however you have to input a name rather than just an email address like Zoopla does. This is the kind of style I was thinking for my website as I believe there should be a name which allows my client to know who he might be seeing. I like the style which the create user page has and also the colour scheme catches the eye and is aesthetically pleasing.



## Features of the proposed solution

|  |  |
| --- | --- |
| **Feature of the Solution** | **How I will solve the Problem** |
| From the interview the client wishes to have an online booking form, which allows user to book. | The way which this is going to be solved is through having a calendar system though ASP.NET which allows you to add a system which my client can add dates onto and time and then this updates on the website meaning that the users once they have created an account they can select the date and time which is convenient to them and this automatically updates making the time slot occupied. The realistic way of doing this will be through a feature on ASP.NET, which allows you to insert a Microsoft Office calendar that you are able to update. |
| A calendar system, which allows the client to track the payments. | This is a different calendar system to that described above as this will go under the heading Calendar. This will be done though the feature of adding a calendar which is linked together through a database, the two will then speak to each other allowing my client to fill in on the database saying when payments are due for each property so when the user logs in they can see all of the dates which might apply to them. This realistically will be one the biggest problems to solve as you will have to have the calendar database talking to the calendar application on the website and then you will have to have this section of the website will then have to talk to the database with the login system allowing the tenant of the properties access to view when their payments are due. |
| The client wants to display photos of the properties and floor layouts. | The way this can be solved is through the ASP.NET there is a section that allows you to add photos and choose the correct heights, width and the place you want them to go this is done through coding the specific height etc. Then you go onto adding to images by saving them into a folder on the sidebar of ASP.NET, after you can then drag the images into the correct place, or if you want them identically on each section then you would use a master page to set out where the image will be placed. This makes it possible to have an image of the property before you click onto each individual property that takes you to a different page that will then have multiple images on. If I have enough time then I will make the photos go around in a continuous slideshow allowing for a more eye catching style making it look more sophisticated. |
| From this the client further wants to have the addresses, prices and description of the properties. | The way which you add the addresses is though linking a database onto the webpage which you want the information. From this you will then need to input on the webpage the specific information that you want pulling from the database for each property, from this you can then edit information on the database and it will automatically update on the webpage. Doing it like this saves time and effort, as you are only needed to change the information on the database rather than having to edit the whole webpage. Also using the database system it means that the webpage will be more identical with the same style font and size for all the information pulled. |
| He also wants to have a navigation menu with the headings: Properties, Calendar, the sign in/ new customer sign up, contact page, FAQ page and a testimonial page. | The way which you would solve this is by using a master page; this master page allows you to create an overall style with how each page will work. Therefore by using the master page it means you can create the navigation bar though both coding to create the correct style along with using a wizard tool to create the basic this webpage links to this navigation menu. Once exited out of the master page it will show the same navigation bar with all the correct styles and fonts on every webpage you click on making it identical. |
| Under the contact page the client wants to have an enquiry’s section, which allows users to send messages to my client. | The way in which this can be solved is by creating the webpage which will be called Enquires, this will then have a web form which is created by drag and drop web form wizard which allows you to add text boxes which you can change to only allow 11 numbers if its for a phone number and will do a query check to make sure in your email address it has an @ symbol and will also compared the two emails which you typed in match each other. It will also involve some code to make sure when the send button is pressed it will send this enquiry to the correct place for my client to receive and respond to. |
| The client wants a classy, friendly colour scheme that is easy on the eye. | The way this will be solved is though using the master page again however this time you would create a file, which you would code in CSS. This coded CSS file will have the size, colour, style and font of the text along with the colour of the background which once this file is dragged onto the master page will make sure that each page will look the same along with the same style of text. This will make sure that it looks classy as the last thing you would want is having five different styles of text as this would make the website look unprofessional. The colour scheme I’m moving towards is a classy black grey or blue with white or black writing depending on the colour of the background. |
| The client finally wants to have a log in and a create user section so users can have accounts on the website. | The way which the website will have a login system is though a webpage to itself which once clicked on will send you to a login / create user page. For the create user this will use text boxes which the user would input their names and email address. This information will be case sensitive. For the password you can set the text box to set the password to a \* symbol once they start typing words into the box which will help with confidentiality and data protection. I will have to do a query check to make sure the email they enter has the @ symbol and also check the email and the re-enter email matches and will have to do the same with the password to make sure that they enter with at least one capital letter and have at least one number or symbol. This information will then be connected to a database that will save the information, which they enter.  For the login system the user would require to enter the email and password, into the web for text boxes, which they entered in the create user section, once they have pressed the login button It will connect to the database and find the email which they typed in and do a validation check to make sure it’s the correct email, it will then use the email to find to corresponding password and do a validation check to make sure that the password is correct to what they entered at user creation. |

## Software requirements

The website in which I will be creating during testing would require a 64 or 32 bit Windows computer with either windows 7, 8 or 10 with .NET Framework 4.5 as these are the only operating systems which support the framework.

During testing the user would require the above software requirements. However if the user wanted to host the website they would have to use a host. Before you host you would have to check that the hosting company can use the Microsoft database which is used in the website. There are several different hosting companies; examples are godaddy.com, 123.com and 1&1.co.uk. The prices of these vary from around 99pence for .online up to 35 pounds for .London.

After the website has been hosted you will require an Internet browser which supports flash for any videos which may be included in the website.

## Hardware requirements

The website wouldn’t require any powerful hardware due to there the website being primarily used on an Internet browser.

The requirements are:

|  |  |
| --- | --- |
| **Component** | **Justification** |
| Processor – 1GHz Minimum | This is the minimum processor speed required for the .NET framework 4.5. Also for an internet browser. |
| RAM – 512MB Minimum | This is the minimum RAM required for the .NET framework 4.5. Around the same required for an internet browser. |
| Disk Space – 4.5GB minimum | This is the required disk space required for the .NET framework 4.5. This disk space is also required to run the internet browser. |
| Mouse | This is required to click the links and explore the areas in the website along with clicking fields within pages of the site. |
| Monitor / Screen | This would be required to view and access the website. |
| Keyboard | This would be used to search for the website along with inputting text into fields. |

(Microsoft Website, n.d.)

## Success criteria

1. The website must have a login system which allows users to create accounts which they can then use to log into the website. This must be user friendly and connect to a database which must store and also recall the correct information once typed onto the web form.
2. From the login system there will also be a create user system which connects to the same database however once the web forms been filled in it must store this information in the database.
3. The login system must also have a query checker which will check the email address against the retype the email section and make sure that they have been filled in correctly with the @ symbol.
4. Finally the login system must have a query checker to check the password, so there will be a re-enter password section which the user will have to type the password in again and this must check against the original to make sure they match.
5. The password system must also have to have a query check for there to be a capital letter, number and a special character of some kind which must allow there to be a strong password.
6. There must be a navigation menu which must be eye catching and easy for the user to read, each of links in the navigation must link to the correct webpage.
7. The navigation menu must have a clear to read text which is clear to anyone who may read the webpage.
8. The colour scheme for this navigation menu must also correspond to the colour scheme for the rest of the webpage to make it look professional.
9. The webpage must show all of the properties which the client is renting, this must be displayed in a smart logical way with images.
10. This must be user friendly and once clicked on each property it must send you to its own webpage.
11. On its own webpage it will connect to a database which must call the information and display the information pulled in the correct text box.
12. The price must be displayed clearly in a different coloured font to make it clear for the user to see quickly.
13. Under each property there must be a cover image which displays an overall image of the property. Once clicked into the property there will be a further collection of images which if there’s time it will turn into a slide show with each image swapping as the main image.
14. The system must have a booking system which the user can access and choose a time slot so they can view the property.
15. This must be done by using a calendar system which my client can edit.
16. The system must have a general enquiry’s form, which allows the user to input data which will then send this information to my client.
17. The website must have a user friendly classy colour scheme which is down to my choice on what I believe will work best.

## Limitations of the proposed solution

One limitation of the proposed solution is some of the features won’t be included due to time restraints which however can be added later on. These features might be members only page which allows users which have a property to access this page, which would require a custom database, which links the login with those separate pages, this would take a lot time aside to implement which could then be brought in later.

Another Limitation would be the calendar system which might be more complicated than first thought of and might turn into a drop down box with dates rather than a full designed calendar which requires a lot of time which might not be realistic in the time frame I have; this could be looked at near the end if I have enough time.

A limitation could also be having to use images from outside the program without them being hard coded into the website. As the problem would be with hard coding is the website would have to be changed manually by a programmer rather than if the pictures where called form the database, which if called from the database would mean that the client could easily change the pictures without having to take the website down to update.

# Design

## Decompose the problem

|  |  |
| --- | --- |
| Problem | Explanation |
| Login Page | This page will have its own section in the navigation menu, this allows users to quickly click and login into the website. Once clicked on the navigation menu they will be diverted to the page which has the create user section along with a login section. This means that it utilises the space on the page along with it being user friendly.   1. From the create user section, there will be a web form which will created through text boxes, these text boxes will link to a database which will allow the administrator, the client, to manage the accounts. 2. From the client having access to the database they can then manage which users have access to certain webpages, for example if the client knows that John is a tenant then the client can then on the database change the privileges of John to allow him access, once logged in, to the webpage which has the calendar, John will then be able to see all the important information for his property. This is a member’s only page. 3. When the user presses ‘create user’ button on the create user section of the login page it does a validation check, which makes sure that the email address they entered is correct and a valid email address. The validation check in this instance will check the email for the ‘@’ symbol, it will also check that the user has actually entered an email address as they could forget to input. The validation check will also check the email they entered against the retype email. The validation check will also check for certain elements in the password, which includes one capital letter and a number, maybe a special character but it depends on how secure the passwords wants to be. 4. Once the above has been completed and the information is able to send by pressing the ‘create user’ button, it will then add the information in a database which will store the information securely. |
| Navigation System | 1. For the navigation system it will use standardised colour, text and spacing. This will be done through CSS which will organise the way in which it is styled and the separation between the each of the bars. The colour which will be used will be in accordance with the original website colours to therefore have an overall streamline look. 2. The navigation system will link together using a pre-built item called sitemap which can be programed into to produce the pages to be linked. The pages will therefore be coded with the correct page names and URL |
| Website Pages | 1. Each of the pages will have the title of what the page is; this will come under the h1 tag. 2. Each page will be created using the Master Page template. This master page stores all the basic design such as the sizes of texts where the main features are going to be. The master page will then be linked to the CSS file. 3. The CSS file will have all of the relevant style along with the way in which each section will fit on the page. The CSS file will have the size of each section along with text size for all the different tags. Furthermore, the CSS allows you to place the images for the menu background, sidebar and also for the header. This CSS file allows every page created to look the same. 4. From this master page a base page is created, this base page is a shared Page class from which all the web pages inherit. This allows all the pages to share common functions and settings. So this means that each page will share the same characteristics. |
| Property Webpage | 1. The property page will be created using the master page inheriting the base page. So it will have the same style as the rest of the pages on the website. 2. A part of the success criteria is that each property will be displayed in a smart logical order with images. The way in which this will work is by having a data base with a table called property. In this table there will be columns with things like property type number of bedroom, bathrooms, price, description. Also there will have to be columns called PropertyID and VenderID, these will be used so that the database can look into each table and find the relevant information relating to data. To then have all of the relevant information to be passed onto the website a query will have to be done looking for the propertyID. On the website an initial first page will have to be created which will use the query to call the information from the table. This will call the address and the price. There will also be a button which will be used as a session which each button will be linked to the propertyID. This session will then be used to pass the propertyID onto a property master page. 3. The property master page will have labels which will use the session passed when the button is clicked at the bottom of the property. The session pass will pass on the propertyID number filling in all of the details from the table in the database relating to the propertyID. 4. In relation to the images, in theory the same method used above should work for images where the image path will get called from the database table and from there the image location can be found and the image should then be able to get generated. |
| Booking System | 1. For this system the way in which I intend to do it is by using a database which will have a list of available times along with available dates, this will then be linked to the propertyID so that when the potential renter wanted to see a property they will click on that property and from this a session will pass with the correct propertyID allowing the available times for that property to be displayed, so that when the person clicks on the calendar on a specific date then all the available times will be displayed which the user of the website can click on the available time slot. Once the available time slot has been selected and clicked on that time is removed from the available times and will be filled in with who is going to view the property. 2. All of this calendar system can be easily edited as it will all be connected to the database table making it possible for the client to edit the information for available times and to also view who is going to view the property. |
| General Enquiry’s form. | 1. For the general enquiry’s form, there will be text boxes which will have to be filled in to allow any information to be sent to my client. The information will include name, email, date of birth, mobile number and then a message. This information must be filled in as there will be a validation check to make sure that all of the information is filled in correctly and there is no information which is missing. The validation check will also look at the email address and the confirm email address to make sure that what they entered matches. 2. From this when they press the submit button at the bottom, if any of the information they have entered is incorrect then the validation check will be run and the information entered wrong will be shown by a red star (\*). 3. Once there are no errors the webpage will be set up so that the information will be sent to my client’s email address. This means that whenever anyone fills out the contact form then the client will get up to date notifications to any questions. This will also be set up so that when the client clicks on the reply button in the email application it will automatically know the email address. This will make it easier for my client to be able to have contact with people having questions about the property. |
| Colour Scheme | 1. The colour scheme which will be implemented for this webpage, will be a grey background colour with the sidebar being a darker colour grey. Also the navigation menu will be a further darker grey to contrast between the logo of the company and the lighter grey of the main content section of the webpage. 2. The text boxes will be the standard white colour. The text on all the pages will be the standard black colour. The buttons will be a blue colour to contrast the grey background. |

## Structure of the solution

|  |
| --- |
| The diagram below, shows the relationship between the way in which the database tables will connect. The relationship between each table are normalised to 3 normal form. This means that there will not be any duplication of data within the tables and that each data inputted will only be displayed once in all the tables therefore cutting down the risk of there being any problems in data being incorrect.  Also, the data tables are all one to many rather than many to many, this is done through having the IDs like ViewerID, AppointmentID etc. placed in more than one table. By doing this it means that only once piece of information needs to be placed in each of the tables, so then it corresponds with information in other tables. This means that for example the Renter table and the Viewerlogin table being linked through the ViewerID that my client will be able to recognise who is a current renter of each property and can see the number so look at the Viewerlogin table and know the name and the rest for that person. This is useful as there doesn’t need to be any repetition of names which could lead to mix ups in the data and the client getting confused. A mix of data could also cause problems with say the login system as if for example they had to retype the email address instead of just using the ViewerID it would mean that other parts of the system wouldn’t be linked to this, therefore could cause massive problems with communicating with other tables.  By breaking down the database into the individual tables shown below and by having a relationship between them it helps with some of the problems which my client wants in the webpage. The first problem this would help with is the login page as it would allow there to be a table which is designed just for the login section, as once people have filled in the information in the create user section of the login page, this information would be filled into the Viewerlogin table of the database using an insert query. From this the user ID can then have access to all of the other tables without having my client having to insert other pieces of information other than the correct user id to have them access to other areas. This clearly is linked as seen below to the renter table, this gives another good example of how relationships will help in the database to solve the problem my client wants solving. As the ViewerID is the secondary key in the renter table meaning that all the information in the Viewerlogin table doesn’t need copying across to the renter table meaning there is less chance of anything breaking. Along with this it also means that one table won’t get too full and make it difficult to manage so by splitting into different tables makes it easy to manage. In the case for the renter table it means that my client can edit and manage who has access to the calendar system by just entering in the ViewerID and checking the current renter box. Also from the database it will make it easier to add information like when payments are due rather than having to enter the website itself and updating the information that way.  By having a table for vender, it means that the venders information can be updated for example if there is more than one person in the business who might organise some of the properties then it’s easy to add them, this then links to the property table and the VenderID can be added to the table so that when managing the database and the business it becomes easier to know which vender is organising and running each of the properties.  The way in which the database will connect to the webpage is queries which will be implemented through the website. Using a dataset in VB.NET will allow easy management of the queries created using SQL statements. The main SQL statements will be select single value or insert, which will allow information to be extracted from each of the tables and insert statements allows information to be inserted into the table. |
| Each of the sections in blue is its own webpage. The ones in the square brackets are all sub categories under these sections. Each sub categories are its own webpage which can be found through hovering the house over the main category on the navigation menu.  Each orange section is the database with table as being the table where the data will come from and the ID is the information which the queries will look for when searching the database.  From the home page, where everyone using the site will first come to, each page is linked with each other to have an effective website. No matter what page you are on you will be able to click on any of the webpages and have access to them.  The Property page will connect to the property master page through a session. All the properties will be displayed with a button which will have the propertyID and then this will be passed on through a session showing all the information which is associated with the property on the master page. This page will change depending on the property clicked on.  The orange line between the properties and the orange box of the table, shows that when the button is clicked on below the property that the information through a query will be sent to the table through the ID as propertyID to recall the correct information regarding the price, address etc. By calling information from a database table it makes it easier for the client to change information rather than having to enter the website itself and editing information within the raw website which can take time and also the website would be down for some time, where using a database the website doesn’t need to be down as the client can update the table and the information will be updated.  The contact us page has a web form which will be created, which includes name, age, email and an enquiry box. This web form then connects through behind text code to an email server. The code should include the email address of the client along with the password to the email account.  This will be useful to the client as sending the query to the email address to my client makes it easier to keep up to date and can therefore be accessed anywhere with an internet connection. The web form also allows there to be easy communication between people and my client making it more efficient than the current way which my client runs the business.  The way which the login system works is that first the users create an account; they do this by filling in the create account sections of the login webpage. Once they have filled in the web form, this information is sent to the database using an insert query this query will take the information from the text boxes and insert the information in the correct column in the table. There is some code which is needed to make sure that all the information is inserted in the correct order.  The way which users will login is by using the email and password they used to set up the account. This information is then used to query the database. The query used is a select which brings the information out of the database, so the select query would then output email addresses and then code would be used to compare the email address which the user inputted when trying to login and any other email address from the database. If the email doesn’t match, then the message box will be outputted with something like email or password is incorrect and the same will happen above with the password. However, if the check between the email and passwords are correct then the message box will display something like welcome to daisy lee holdings.  For the calendar system, the way in which this will work is once they have logged in through the web form they are directed to the calendar webpage under the about section of the navigation system. From this information, the ViewerID will be passed from the web form login, the query used will look for the ViewerID and will also output if they are a current renter. Because current renter will be a tick box it means that the output will be true or false. This means that if in the database table renter has the id of those who are renters of ex renters or even those who might have a chance in renting then they won’t have access the calendar system as they won’t be selected as true in the database therefore the system and won’t be able to see the important dates. From this also as shown by the grey message box it will have a pop up message telling anyone who tries to access the calendar by clicking on it that they aren’t a current renter therefore not gaining access to this system.  To follow on from this once the message box has been displayed and the user of the website has clicked okay on the message box there will be code on the webpage so that they are redirected to the login page.  However, if the user is a current renter and in the database, the tick box is ticked and therefore in the code their ViewerID in the renter table will match with the Viewerlogin data which when they login the data will be passed by a session and using the query explained above to pass the information on. Therefore, the information will be true and they will gain access to the information within the calendar section of the webpage.  For the reviews section of the webpage there are two sections which the user can click on when they hover over the reviews section either by property or by all reviews. Both ways will work in a similar way.  For the by property reviews this will work by having a web form in terms of either a drop down box or a set of radio buttons. So when the user selects the property a query will be involved so that it selects the property chosen and using code the query will know which the propertyID is for each property. The query will be a select as it will select a single row outputting the information for that property which would have been inserted into the database.  For the All reviews it will do the same process but instead of having to select a property it will just output all of the reviews. So it will do a select query and will output everything which is in that database table. Which is why there doesn’t need to be anything going through the web form. |

## Algorithms

Contact Us Page

1. IF Page is valid THEN
2. Mailbody = mail body replace (##Name##, with txtname.text)
3. Mailbody = mail body replace (##Email##, with EmailAddress.text)
4. Mailbody = mail body replace (##Mobile##, with mobile.text)
5. Mailbody = mail body replace (##comments##, with comments.text)
6. Message subject = “Response from website”
7. My message body = mailbody
8. Message sent paragraph.visible = true
9. Contact form .visible = false
10. End if

Property First Page

1. PageLOAD
2. Property label = tableadapter Selectaddress(3)
3. Price label = tableadapter selectprice(3)
4. Property label2 = tableadapter selectaddress(4)
5. Price label 2 = tableadapter selectprice (4)
6. When property button 1 pressed
7. propertyID 3 = session called PropertyID
8. Redirect to Property Master Page
9. When property button 2 pressed
10. propertyID 4 = session called PropertyID
11. Redirect to property master page.

Property Master Page

1. When pages loads
2. Session = PropertyID
3. Session = RenterID
4. address label = tableadapter selectaddress(PropertyID)
5. bathroom label = table adapter selectbathroomnumber(PropertyID)
6. bedroom label = table adapter selectbathroomnumber(PropertyID)
7. description label = table adapter selectdescription(PropertyID)
8. price label = table adapter selectprice(PropertyID)
9. type label = table adapter selectPropertytype(PropertyID)
10. label description 1 = table adapter selectdescriptionone(PropertyID)
11. label description 2 = table adapter selectdescriptiontwo(PropertyID)
12. IF RenterID = 2 then
13. Upload button visible = True
14. Data grid view visible = True
15. File upload visible = True
16. End if

Login Page

1. Create User Button pressed
2. IF page is valid then

Insert as new login viewer into ViewerLogin table (txt\_forename.Text, txt\_surname.Text, txt\_dob.Text, txt\_email.Text, txt\_createpass.Text, False)

1. End if
2. Login button clicked
3. r = Login select (txt\_loginemail.Text, txt\_password.Text)
4. IF r is nothing then
5. message box (“Invalid Email or Password”)
6. else
7. message box (“Welcome to daisy lee holdings”)
8. Session = RenterID
9. RenterID = SelectViewerIDFromEmailandPassword(txt\_loginemail.Text, txt\_password.Text)
10. Redirect to calandar page
11. End if

Calendar Page

1. Page Load
2. RenterID = session
3. IF SelectCurrentRenterFromRenterID(renterID) = True Then
4. Property rent label visible = true
5. Property rent label = selectDateOfPaymentsFromViewerID(renterID)
6. Else
7. Message box (“You are not a current renter")
8. Redirect login page
9. End if

## Usability features

|  |
| --- |
| One of the main features that will be the main part of the webpages is text. The text will be implemented by having a CSS file which will dictate the style of texts along with the colour. For example, in the CSS file there will be a section which called ‘font-family’ this will define the main font used for the entire website. Therefore, this will make a standard look for the website, the style which will be used is Arial, Sans-Serif which will allow ease of read.  From this there will be a further section in the CSS file which will make the h1 (heading one) titles which will make it smaller than a browser default, the code which will be used is ‘font-size’ this will then dictate the size of the text, the size of text which h1 will be throughout the website is 20px. For the h2 (heading two) element of the webpage will be done near enough the same but will be dictated by the h2 tag. The h2 element within the website will have font size 14px and colour black.  By having the text black on the website, it means that the colour will contrast against the light grey background of the main content section of each page, by doing this it will make sure that users of the website are able to read the text displayed on each of the pages. |
| Another Usability feature is the buttons on the webpage, this will be a key part of the website due to buttons being one of the key parts in making the website run as it should. Therefore, for each button the CSS file will dictate the colour of the text boxes along with the colour of the text displayed. The colour of the text boxes will be a blue colour which is useful as it means they contrast the grey of the background and makes it clearly stand out from the rest of the page. By doing this it makes it clear that this is important and is a key part of the website to make it function, therefore making it obvious that they need to be clicked if they want to get somewhere with the page. Also, the CSS file must change text inside the buttons to white, this makes it useful as white clearly stands out from this white making it easy to read. |
| A useful usability feature which is going to be included in the website is the use of message boxes. You may think that they aren’t a useful usability feature however it is a very easy way for important information to be passed from the website to the users of the page. Therefore, say for the login page when they are attempting to log in and they enter an invalid user name or password then a message box will be outputted and will inform the user trying to login that something is invalid. By having a message box popping up on screen it makes it obvious that there is an error or piece of information which is important. Also from this by having a short piece of texts on the message box both for the title of the message box and having a message in the text box it makes it easier for the user to know what has happened. |
| Validation checks is another useful usability feature as it will allow users to see, for both the create user section and the contact us page, the errors which have taken place. This is done through checking that required fields have been filled in and that they meet the standards for the system to work. For example, a validation check would be that the email and confirm email are of the same, also to check that they include the ‘@’ symbol somewhere in the email address. The reason why this is a usability feature is for example somebody who maybe not used to online forms etc. then by having validation checks it makes sure that they are entering the information correctly and will also display anything which may be entered wrong in a red colour to make it stand out from the rest of the text. The red message will inform the user that information has been entered wrong and how to correct the mistake. |
| The final usability feature is the use of labels. These are useful for having information being placed into them. These can be used when information from the database is being called from, so information can be placed within the label making it easier when coding which piece of data goes into each label. The benefit of using labels is that you can edit the size of each individual label so the text size and the font can be changed on an individual basis. This is a usability feature as it means important information can be changed, so this might include making this specific text bigger or even changing the colour making it obvious to the user which information is of most importance to them. |

## Key variables and structures

Inherits BasePage – This will be used on every single page as it will inherit the basepage class

Login Page

Public loginViewer – This will be used as the variable to store the viewerloginadapter

Protected Sub But\_createuser\_Click – This will be used for when the user presses the create user button on the login page.

Private Sub But\_login\_Click – This will be used for when the user presses the login button on the login page.

Dim r – This will be used to store the email and password and input them into a query which will compare what the user has inputted and whats stored in the database. From this the user, will then either be given a message saying invalid email and password or If not then they will get a welcome message.

PropertyMasterPage

Global Variables

Dim tableAdapter – stored information from propertytableadapter made in the dataset. This can then be used to output the queries created.

Dim list As List – This will then be used to store the information from the dynamic list and will be used to add extra elements. This means that more images can be dynamically added to the array.

Dim arrayFromDGV() As String -This will used to make the array into a string.

Dim tableAdapter2 -This will be used to store the Picturetableadapter so in the code it can be called upon to add the queries created in the dataset.

Local Variables.

Dim PropertyID – This will be used to store the session which has been passed from the first property page. This will be used to store the propertyID which is associated with the button which the user has to press to get to the propertymasterpage. The queries can then use this to get the correct information from the database.

Dim RenterID – this will be used to store the session passed from the RenterID so that the admin can edit the page.

Dim arraySize – This will be used to count the number of rows in the data grid view.

ReDim arrayFromDGV(arraySize) – This will be used to redim from the data grid view to set the size of the array.

Dim Pics – This will be used to set up the picture element as a set of list.

Dim imageString(3) – This will store an array for the images and will only beable to add a max of 4 images in the array.

Dim url – This will store the information from the data grid view and will begin the array

Protected Sub btn\_upload\_Click – This will be used when the user clicks on the upload button

Dim pather – This will be used to store the full path of the image.

Dim pathSplit() - This store the use split and delimit by \ to break up path

Dim pathSplitSize – This will be used to find file name in the element in the array

Dim sFileName , Dim fileType – This will be used set the upload file name.

Dim saveimage – This will be used to save the file to the disk.

First Property Page

Global Variables

Public tableAdapter – Will make the propertytableadapter available to use the queries for the webpage. Makes the table adapter from the dataset visible for the functions to be used.

Public image -Will make the picturetableadapter visible so that other functions on the page can access the dataset to use the queries created.

Local Variables

Protected Sub but\_propweb1\_Click – This will be used for when the user presses the property 1 button then it does the code under this section.

Session("PropertyID") – This will be used to stores the propertyID. It stores the information as a session so can be passed between pages.

Response.Redirect – This will redirect the user to the propertymasterpage.

Protected Sub but\_propweb2\_Click – This will be used for when the user clicks on the property 2 button.

Contact Form

Protected Sub SendButton\_Click – This will be used for when the send button is clicked in the contact form. It will then run the code which is in this sub.

Dim mailBody – This will be used to read text file for the email. The reason why its called mail body is because it reads the text file which makes the body of the email and then it will be used to store the replacements to makes the email body so they are all identical.

Dim myMessage – This will be used to create a new email. Therefore, it is called my message as it will store the message which I want for the email.

Calendar page

Public currentrenter – This will be used to store the rentertableadapter which means that in the program other subs can access the queries created in the renter table.

Response.Redirect – This will be used in this context if the user that has logged isn’t a current renter therefore will be redirected back to the login page.

## Test data for development

Login Page

For the login page, the way which I will test the different functions are: For the create user section of the page I will input ‘normal’ data which will include a name, data of birth and email in full with the correct symbols included. Once this is complete and I am sure that this works correctly therefore I will check that all this information has been correctly inputted into the database. I will then try something anomalous which will test the validation features. The reason for this is to make sure that the validation checks work correctly as if they don’t work with the correct data then there is clearly something wrong.

To test the validation features I won’t enter any information at first and just click the submit button and test that the validation works and nothing gets inputted into the database. Once this has been done I will then go down the list inputting one piece of information at a time then click submit and makes sure that the correct validation rules are being displayed and nothing is still being inputted into the database. Once this has got down to the input email section of the table I will then input an email with no ‘@’ symbol and leave the confirm email blank, click submit and check the validation rules are being displayed and nothings in the database. I will then for the confirm email address just use a different email address and check the validation rule compares the two. Once this has been completed I will enter correct email addresses with the ‘@’ symbol and make the emails match. I will then click submit and check that the validation works and recognizes that the emails match, there still shouldn’t be anything in the database. The next stage of checking the create user page is by entering in a password in the password text box and then in the confirm password text box I will then use a password which doesn’t match. I will then click submit and check the validation compares the two and output a message saying they don’t match, also will need to check that no information has been inputted into the database. When this has been checked, I will then input two passwords which match and this should then be accepted by the validation. I will then click submit and there shouldn’t be any validation errors being reported due to everything being correct, I will then check the database table Viewerlogin which should have all the information from this test. This will show that the query created to input the data into the table works correctly therefore this will be a test for both the query and the validation. The reason why this is required is to basically double checks that the validation which was created works as it should, because any data which is invalid then creates problems with calling information from the database and for my client to understand why information isn’t being entered correctly.

The next stage for testing the login page is to test the actual login section, for this I will use the user which I would have created in the create user section. I will then input the wrong email but the correct password and then click login and check that an invalid message appears. Then I will input the correct email but the wrong password, this should still come with an invalid message as it still won’t be the correct information which is stored in the database. This tests the select query which will be used for this therefore if the user can login with the wrong information then it shows that there is a problem with the query created. I will then input the correct email and the correct password which should then log me in and show a welcome message, so will also show the query works correctly. This test is required to make sure that the login works and the correct information is called allowing users to then be able to login without a problem.

The next stage of testing will be once logged in then the user if a current renter will be sent to the calendar page as in the database they will have a tick in the table renter. If they aren’t a current renter, then they should receive a message saying they aren’t a current renter. To test this I will firstly in the database make sure that the login details I have just used in the renter table is unticked therefore shouldn’t gain access and receive an information message. I will then in the database tick the box and will therefore gain access to the calendar page. This will show that the query to the information works correctly how I should, also check the message boxes are all correct and lastly checks that the user is direct to the correct page. This is required as it will make sure that the correct information is being called from the database so that only certain users are able to get access to certain pages.

Calendar Page

For this page the way which I will test if the database and website connect with each other and output the correct information. I will change the ViewerID to that of the account I created above and change the date to the 13th March 2017. I will then load the page and check that this is showing the correct information. Once done this I will then change the date to the 14th March 2017 and then refresh the page and make sure that the date reads correct therefore shows that the database is communicating the website.

Property First Page

The way which I will test that the webpage and the database communicates correctly using the queries is through in the table property change all the information so that it reads what the title is therefore I can check that all the information is in the correct place and nothing’s going in the wrong order. Once this has been done then I can change the information back to text of what the property is etc.

Property Master Page.

I will use the same idea as what’s described above to make sure that the information is in the correct order for this page. I will also have to check that the buttons on the Property first page work correctly and display the correct information regarding that property. Therefore, I should use the numbers 1 and 2 at the end of each heading to make sure that this information is being displayed with the correct property information.

I will also have to check that the images are working correctly and displaying the correct order and the correct images for that property. Therefore, to test this I will get images with numbers on ranging from 1 to 8. Images 1 to 4 will be used for the first property and 5-8 will be used on the second property making sure that the images displaying correctly.

Then for the upload I must login with the admin account and then I will click on the first property, upload an image of a red apple and check that it is correctly added to the database then will replace the propertyID to 3. Then for the second property I will upload a green apple, replace the propertyID to a 4. Once this is complete I will refresh the webpages and then the two images should be displayed showing that this works correctly.

Contact Us.

This page will be done in a very similar way to the create user due to the validation working in the same way. To start with I won’t enter any information into any of the text boxes and click submit. Once I’ve done this I will fill in the first text box which is name and then I will click on submit and see if the validation works correctly. If it doesn’t then the information will be sent by email to my client. The second stage is to fill in the next text box which is email address, I will use an email which is incorrect and doesn’t have the correct symbols for an email to work then I will click send and an error validation message should return saying it is incorrect. Then for the next step is to check the confirm email to do this I will use a different email address than the one used above, this will check to make sure that the validation that compares the email and confirm email works correctly. When I know that this works I will then use the same email in the confirm email and then will click send, this shouldn’t bring an error message for anything but mobile and comments. I will then input a mobile number which is 11 numbers and then this shouldn’t bring a validation errors. The next stage is comments I will input some text ‘TEST’ and after I will click send.

Since there will be no validation errors the information should then be sent to the email address which will be set up in the website. To test this I will login into the email account and check that the email has all the correct information which was inputted into the textboxes.

## Test data for beta testing

Login Page

The way which I will get the user to test the create user section of the login page is by making them enter information into the text boxes like how they usually would. They will enter information by what it says next to the text boxes, so if it says ‘Forename’ then they will enter their first name etc. This will test to make sure that the webpage isn’t over complicated and they can make sense of what information that they need to enter. If this is successfully entered and the user successful creates an account, then it means that this test is successful however if an error occurs then it shows that there could be too little information for the user to know what needs to be inputted.

The next test which will need to be tested is the actual login section of the page. The admin user of this page will firstly have to choose which user has access to the calendar page by filling in the renter table and making sure that current renter is selected and the ViewerID has been filled in. The user can then login. This will test if the error message works correctly and if the admin user has correctly entered them into the database it will check that the welcome message works and the queries work correctly pulling both the information in regards to when the payments are due also the renter information is correct.

Property First Page

The way which the user will test this page is by pressing the property buttons and making sure that the property which they clicked on takes them to this information therefore showing if the buttons work correctly.

Contact Us Page

The user can test this page by inserting information into each of the text boxes. They will insert the information as if they were acting trying to send an enquiry to my client. By doing it this way it should just check that the information which they need to enter is clear it should also check that the information sends correctly to the email account. However, if this information does not send and there are validation errors then it shows that either the user has incorrectly entered information which means that I need to maybe change the text to make It more clear what needs inputting.

The admin user will need to test that they can login into the email account and they are seeing the correct information which has been entered, they will also need to test that when they click the reply button then the email address which the user entered on the contact us page will appear in the to section of the email message.

# Iterative developing of the coded solution

## Iterative development

|  |  |
| --- | --- |
| To begin the website, I created a new website by clicking on file, new, website. This then creates the new website page as shown on the right. At the beginning, it doesn’t include the Frontend.Master. To create this, I right clicked on the WebSite, click add then add new item. A list of items appears and I clicked on Master Page, from this I renamed the file to Frontend.Master to make sure that I knew what that specific page is for. This page is mainly for the basic outline for how each page will look like. At the moment, this page will be blank. |  |
| The next stage of developing this website was to add a style sheet. To do this as shown on the right is to add a new folder called theme this is so that further down the line when there a more files that you know what the folder contains. Because it’s called theme it becomes obvious that inside of this folder contains the basic theme of what the page will look like. The title of the Theme folder is monochrome as it easily and quickly tells the developer that this is where all the information is regarding the theme of the website. |  |
| This shows how to add the style sheet which is where the CSS will be implemented to make the website look how it should and to make sure that each page will look the same with the correct text and colours etc. This file will be placed in the folder as shown being created above. The Style Sheet will be called monochrome; this is a fitting name for the style of colours which will be used for the website. This is because the colours being used a varying shades of grey. |  |
| On the right it is showing how to add a skin file, a skin file is basically a button style sheet which uses CSS. By using a skin file, it allows you to define a look and feel of each button. By placing it in the Monochrome folder it will change all of the buttons to how they have been designed in this file. By having a .skin extension it contains all the serve-side presentational elements of a control, so the settings are applied to all the controls which the skin applies to. To see the code implemented for the Skin file see the Annotated modular code. |  |
| This part of development shows the images which will be used to create the website design. All in which have been created to make them have the right shape and size for the website. All of the images have the same colour scheme as what the website will have to make sure that there is a consistent classy look. For the logo I had to create my own as the business doesn’t have its own distinct logo therefore I created a basic looking one which my client signed off as being fine to use. |  |
| This is the CSS style sheet which I showed how to create the file earlier on. The screen shot only shows the first section of the code which is used, to see the whole of the code see the annotated modular code. The CSS code creates all of the design in which the website is going to do. By using CSS it means that each page will look the same as it defines how each of the text will look along with size. It also creates the margins between the navigation sections to give a consistent look. The CSS also gives the page the colours, therefore this creates the ‘monochrome’ look, so makes the menu a darker grey from the background, etc.  The CSS has also been programmed so that if we set in properties something to have a CSSClass of a form to ErrorMessage then it makes the text red.  The CSS is vital for the website as it creates the margins in which the page sits in, so will set the total width for the entire page. |  |
| This shows some of the HTML code on the master page ‘Frontend.master’ so that you are able to define in which section the relevant images are going to go. The id= “ ” refers back the CSS code created earlier. This means that the <div> section will have the CSS class defined as PageWrapper.  The <Div> tags are used to group block-elements so that they can be formatted by CSS.  The text like ‘Header goes here’ are only temporary to check that everything is in the correct place before the images shown below are coded into the CSS. |  |
| This shows the website once the CSS file had been updated to include the images shown earlier. The way which the images were added was by background-image: url(Images/Header.jpg); The above code for which says where the images is in the Solution Explorer. Once all the images where added I dragged the CSS file onto the master page. This meant that everything which was coded for in both HTML and CSS this then changed how the website looked from being a plain white document into the desired design I created. The image on the right is a screenshot for how the website looks in the VB.NET console but the desire look will continue through onto how it would look on the internet. The desired look will stay the same however there will be slight altercations depending what is placed on each page. |  |
| This is selecting a web form so that there is a main section in which text and other things like that can only fit in this section to make sure that other areas of the webpage cannot be edited. On the right hand corner I had to select ‘select master page’ this makes sure that the web form uses the master page. |  |
| In this screenshot it shows selecting the Frontend.master as the place to add the webform. |  |
| This screenshot shows that the area in purple is the only place that you can type on the webpage. This means nothing can accidently be changed on the webpage which could cause massive disruptions and delays.  This area is coded in the CSS which was created earlier as everything is defined in there and is called section#MainContent. This means that the main content area can be changed at any point. In the code is defines how big the area is and the minimum height so that even when there’s nothing on the page then each page will still have a consistent style. |  |
| The next stage once all of the above worked I created a new folder as shown on the right, the folder was called app\_code. An app code folder is used as it automatically compiles at run time. The use of the app code folder is also that the code within the folder doesn’t need to be compiled it can have source code in. They are useful as it means I am able to create custom classes and other source code only files and use them in the web application without having to compile them independently. |  |
| The next stage once created the app\_code folder is to right click on the folder and add a new item. Once done that I searched for class and named the class as BasePage. The use of the base page is to use it as a class so that it inherits the information which I set it to rather than automatically inheriting from the page class in System.Web.UI. This gives the behaviour required for each web page to be requested by the browser and be processed by the server. |  |
| This code is then inserted into the base page so that each page created has to have its own title if not then an error will appear saying the page cannot have an empty name. This is then saved. |  |
| Front the base page created above we save this as MyBasePage so that when I create other pages then I can select this page and it will be both the master page and the new base page class which was create above, therefore each page is exactly as the pervious other than the content which will be placed on each page. This makes sure that there is a consistent look. |  |
| After all of the above has been completed I came to a finished design which is shown on the right. This screenshot shows the webpage running in a browser which is running correctly as it uses the base page class along with the design from the master page. |  |
| This is the button.skin file which I showed earlier on in the screenshots. It’s used to add CSS to all the buttons in the webpage to give them a consistent look. From this code you can see that it uses the CSS class which was defined in the monochrome.css file also shows that the back colour is blue and forecolour is black which will give it the distinct colour to stand away from the background, which is one of the usability features. |  |
| This a button which will be created from the above button skin. As you can see it creates the blue with the white writing which makes it easy to stand out from both the grey background and the background of the blue button. These buttons will be used whenever something needs to be submitted. |  |
| The next stage of creating the website was to add new folders as shown on the right how to do. I added a new folder for each of the main sections of the webpage.  The reason why I used folders is to keep the solution bar tidy and make it easy to know what is in each folder so that if somebody else took over they would easily be able to tell what each folder contains. In each folder will sit the actual webpages along with any new folders which may need to be created like for images. |  |
| This is the screenshot showing what the solution explorer looked like once I had created all of the relevant folders. The next stage in development is populating each folder with the webpages associated with the folder. |  |
| This is beginning to show how I created the webpages. Earlier on in the screenshots I showed that I saved the master page with the base page class. Now is the time which I will used the MyBasePage. To get to this point I right clicked on a folder which the webpage needs to go. I then clicked on add, then add new item. From there is came up with the screenshot as shown in the right. I then clicked on MyBasePage, gave it a sensible name as show AboutUs to know that this page will include information about the business. |  |
| I then needed to repeat the process as described above to add each of the webpages shown on the right. These make up the websites for the about section of the website. With the default page always being the navigation bar choice. Which you will see later. |  |
| Each page needs to have a title, as shown on the right, if it does not then the page won’t work because of the base page which we set up earlier. Each page which inherits the base page class needs to have a title on the top row on the source of the webpage. It must go in the correct place shown otherwise there will be an error which occur. |  |
| This screenshot shows all of the added webpages into the correct folder using the methods described above. Each page which was added has a title included to prevent any errors for occurring when the website is running. |  |
| The next stage of developing the website is to add a navigation system to allow access around the website. To do this I used a Site Map, which I found by right clicking on WebSite2 clicking on add, then add new item. I then searched for Site Map and once found I clicked add. This was then added to the solution explorer ready to be edited. |  |
| In this stage of creating the navigation system I opened the Site Map by double clicking on it in the solution explorer.  I then added the code as shown on the right and will be further shown in the section Annotated Modular Code. The code is used for when we add the navigation menu further on in the development process. The code just points towards where in the solution explorer each page is and then lets you add a title so in the browser in the url it gives it the title which I wrote down. The description is used so when you hover over each section in the navigation menu it gives the description to let the user know in a small amount of detail what is on the page. The navigation menu itself will have to be created next. |  |
| The next stage is to add the menu itself. To do this I went back onto the master page and dragged the menu which is under the heading navigation in the Toolbox. Once that has been dragged in I clicked on the arrow on the right hand side and then for choose data source I clicked <New data source>. From this a new window popped up as shown on the right saying to choose a data source and because I created the Site Map as shown above I clicked on that as it would inherit everything which I created earlier. Then I clicked on ok. |  |
| The next stage of creating the navigation menu was to edit some properties. On the navigation menu itself I right clicked and selected properties. In properties I had to change the following on the right to select false and the orientation to be horizontal, this meant that the navigation text would go across the page rather than down the page. |  |
| Once the settings had been changed I had to click source on the master page and find the <nav> tags. In the <nav> tags I had to add as shown of the right the CSS class. This meant that the monochrome CSS file which I created earlier the navigation menu could change to how I wanted it to fit. |  |
| The next stage was to change a property in the SiteMapDataSource. To do this I right clicked on the above and clicked properties. In properties I had to change showstartingnode to false to make sure that the data displays horizontal along with the drop down menus rather than going vertically down the page. |  |
| This shows the finished navigation menu, using as described above to create the menu it has subsections for each page and also has the title and descriptions added. |  |
| The next stage in developing the website was to develop the contact us page which is a key part of the webpage. To begin this, I created a new folder called Controls. Once the folder was created I right clicked on the folder clicked add, then add new item. The menu appeared and I created for ‘user’ this then came out with the web user control, I changed the name to ContactForm and clicked add. |  |
| The next stage of creating the contract us page is in the web user control I just created is to open it up and add a table then add the information shown on the right. I just added a text box for each of the sections and renamed the textbox id to for example txtname or txtemail so that later on I know what they are called. |  |
| The next stage was to add some validation, to do this form the toolbox I dragged a required field validator next to each of the text boxes. Then I right clicked the validator and changed the settings to each one to like the one on the right. However, depending on the text box it was next to I had to change control to validate to the correct text box and also change the error message depending on what the text box was supposed to show. |  |
| This validation was dragged in form the toolbar under the same heading Validation. But instead the validation used is RegularExpressionValidator only used for the email address text boxes. This is because this validation checks to make sure that the email used is valid by checking it has the correct symbols as seen on the right. If the email is found not to have the correct symbols for an email address, then it will display the error message of ‘enter a valid email address’ |  |
| This type of validation goes next to the confirm email address. This validation is a compare. As shown on the right it compares the email address and the confirm email address to make sure that they are correct and match eachother, if they do not match then an error message is displayed telling the user that they need to try again. |  |
| This section is the end of the contract page design, this is the end message at the bottom of the page. This is just a helpful message to let the user know that their message has been sent. The properties of the message have been edited so that it isn’t visible while the user is filling in the contact form but once they click send and the validation is fine then it will appear, you will see how this will be in a step further in the development |  |
| This stage of development of the contact page is behind the scenes where you don’t particularly see it on the webpage itself, however it is one of the most important parts of the contact us webpage. The code on the right as seen shows in the tags <mailSettings> basically this connects to the client’s email address to send the information filled out in the contact sheet to them. The code is needed as it needs to access the email account therefore need the email address the host of the email address and the password for the account. Above that is the ‘from’ which will say the message has been sent from the website Daisy Lee Holdings so in the email that goes to the email account it will display this. |  |
| The next stage to get the information sent to the email account was to add a new text file. To do this I right clicked on the app\_data folder which I created and then selected add then add new item and searched for text and then selected text file. I then renamed the file ContactForm. |  |
| The next stage was to populate the text folder with the information as shown on the right. This will be used as a template for the information in which the user fills in will then populate the template and then the information will go in the format to the email address. |  |
| The code on the right will be explained further in the annotated modular code section. But basically what the code does in the mailbody it uses the file we created above and then replaces the information in the ‘##’ with the contents of what is in each textbox.  The code that is then below creates a new email and adds a subject of ‘response from website’ and then imports the mail body creates above.  The code below that then fills in the from and fills it in with the sender name and then adds the receive name to the message. Then when you click reply in the email it will automatically populate the email address with the email the user left in the email box.  The HTML code which is displayed below shows how the contact form gets imported into the default webpage which is displayed in the website. |  |
| This screenshot shows the finished contact page running in a browser. It shows all of the required features needed for the contact page and this should run without any errors. See the testing section to see if it works correctly as it should. |  |
| The stage in developing is adding a database. This is needed for both the login page and the property pages. First you need to add a database. How you do this is by clicking where the red ring is round on the right. Once clicked on that a new window opens asking you to browse for the database, then I looked for the database which I created earlier. |  |
| Once I had added the database to the server explorer I could then add a dataset. The dataset is where all the queries will be stored. To add a dataset, I right clicked on the folder app\_code clicked on add then add new item. I then searched for data and clicked on dataset and kept the name it came with which is dataset2. I then added to the solution explorer. |  |
| Once in the server explorer I then had to drag over each of the tables into the dataset, once that had been completed it looked like the screenshot on the right. The lines in between the tables show the relationship between them, these were created back in the access database when I was first creating all of the tables. |  |
| This is the login page; this was created on the actual login page rather than on a separate file like the contact us form. However, this was created in a very similar was as it uses a table to get everything in the right place and uses text boxes dragged in from the toolbox and they have also had their id edited to make it easier to code. So the id for login email textbox is txt\_loginemail and they will all go along those lines so for password its txt\_password but for the create user there are text boxes with names like txt\_createconfirmpass for the confirm password. These names make it easier to know what textbox is what.  The buttons as you can see are also dragged in form the toolbox and also have either id names changed to make it easier to code but they still keep the distinct blue colour as created earlier. |  |
| The next stage once all the text boxes where in and names changed along with the database in place, was to add a query so that the create user page communicates with the database. The type of query used is an insert query, this is because all the information from the create user section needs to be inserted into the database in order for the login system to work. |  |
| This SQL statement on the right shows how I added the information from the tables into the database. From this you can see that you insert all of the information from the textboxes into the table Viewerlogin, all of the information needs to be in order of the text boxes otherwise the information won’t be inserted in the correct way in the database. Then the values are all question marks because we don’t know what the information is until the user types it in. |  |
| I then needed to save the query that was created above. Its best to save queries as a sensible name to know what they do for example this query inserts new information into the table Viewerlogin therefore is names InsertNewViewerLogin. |  |
| The code on the right was then needed. You will be able to find the code easier in annotated modular code later on in the document. This code runs when the create user button is pressed. And if the page is valid once the validation checks have been completed and there are no errors then the query we created above runs and everything gets inserted into the database. The validation in this code checks if the email and confirm email address match and if they do then the code can run but if they don’t then a pop up message box returns saying the email addresses don’t match and then the same with the password but instead if the password doesn’t match then a pop up message box appears saying the passwords don’t match.  The code below that basically states each text box and clears the contents of the information once it has successfully passed into the database. |  |
| On the right is a select SQL statement which is used for the login part of login webpage. This SQL statement will select email address and password and count how many times it tries. It gets this information from the Viewerlogin table. It will count once so only goes in one cycle and will look for the email address inputted and the password they entered to make sure that they match. |  |
| This query is then saved so it can be called on in the code which will be showed later on in the development. The name it is saved as it LoginSelect as it selects the login information for the database. |  |
| This is the next query which is implemented. This is also a select query so will select the ViewerID from the database and output this but will also select email address and password but won’t output. |  |
| This shows the name of the query which has been given a sensible name ‘SelectViewerIDFromEmailandPassword’  As this explains what the query does so when it is placed in the code it makes it easier to know what the section of code is called. |  |
| This code shows when the login button is pressed then it uses the query LoginSelect to get the information from the database table Viewerlogin it then it looks at the email address and password and then compares the information to see if they match and if the result comes back with nothing then a message box returns with invalid email or password. However, if it returns with information then it will display the message box welcome to daisy lee holdings.  This code also shows a session which passes information from one page to the next. The session is called in this page ‘RenterID’ and it gets this information from the query which I created earlier called ‘SelectViewerIDFromEmailandPassword’ and gets this information from email and password which they enter. Once the query has this information it will output the RenterID which is in the database. Once this information has all been found the page is then redirected to the calendar page. This will be shown later in the development. |  |
| These screenshots show an SQL select statement, what it shows is the basic idea for how each of the below statements will be selected. The SQL shows the address from the propertyID. This information comes from the table Property and the information for each property will be searched for using the propertyID.  Each of the screenshots below show how the SQL statement can be manipulated to change the Select to which information is needed. This is shown by selecting all of the information from the database. |  |
| The next stage of developing the property page, is to open the properties folder and selecting the default page. Once the default page was selected I dragged in the labels from the toolbar and changed the properties so that the ID was that shown in the square brackets and removed any text. This was so that the information which I selected could go inside the labels, this will be shown in the development stage below. I then also dragged in from the toolbox the button and changed the properties to change the text to ‘Click to view property’. |  |
| This code shows the for each of the labels which I created above that the queries which I created earlier in the development above that they link depending on what I want the information to show. For example, for lbl\_prop1.text = tableadapter.select\_address(3) this shows that in label property1 that the query from select address is used and will select the information from the PropertyID depending on the number at the end. So the example above will output information form the PropertyID 3.  The code also shows that for each of the buttons which are pressed named button id as ‘but\_propweb1’ or ‘but\_propweb2’ this means that depending on which button is pressed then it will pass a session called propertyID which equals the information which is in the labels. So for example the session for but property1 will pass information for PropertyID of 3. Also when the buttons are pressed it will redirect to the PropertyMasterPage which will be explained below. |  |
| The next step was to create a new webpage and I called this page PropertyMasterPage. This is because it will be the basic template for each of the properties. I then dragged in labels for the information to sit in. I changed the properties to remove the text and change the label ID to what you can see to the right. |  |
| The information from the session above is then received on this page and is given the variable of PropertyID. The information to fill the labels comes from the queries which I created above. The PropertyID gets into this page by the session which I created above. The session carries the propertyID from that button into the page and then it will load that information depending on the ID which gets brought into the page. |  |
| This screenshot shows the information which is in the database in the Property table. It shows a populated table which the information will then be passed into the webpage. You can see that the PropertyID is 3 or 4 depending on which row which the information is on. |  |
| This is the opening page which the user will come to when they click on properties in the navigation bar. It gives the user a choice to decide which property they want to see in more detail. When they click on the button of the property they want to see more each button has the PropertyID of the information which is in the labels. This PropertyID is then stored as a session, then the user gets redirected to the PropertyMasterPage. |  |
| This is the PropertyMasterPage which has been populated with the PropertyID which has been passed from the first page by the session, this then uses the queries which was created earlier to call the information from the database into the page using the ID from the session. This therefore makes it easier for the client to change the information in the database instead of having to enter the website and taking the website offline to update. Therefore, doing it this way makes everything more efficient. |  |
| This code shows the variables that will be used to attempt to add images into each of the webpages. These variables are global variables so will be used in more than one of the functions. |  |
| This code follows on from the above code. This code is placed in the PropertyMasterPage. What this code does is create an array from the URLs which are stored in the data grid view on the website page. The way which I got the data grid view (DGV) onto the page was opening the server explorer finding the picture table and then dragging the column called PictureURL.  The code then creates an array, it adds each element from the list to an array, once its done that then it adds text to populate the array.  The bottom half of the code creates the image from the array and then adds it to the webpage.  For more detail on what individual pieces of code do see annotated modular code later on in the document. |  |
| Before I did any of this code a dragged over from the toolbox fileUpload. This is used so that a user can click on browse and can find and select a file which they want to upload. I then dragged a button from the toolbox and edited the properties so that the text was ‘Upload’ and the ID was but\_upload. The code basically checks that the file is valid then it gets the full path of where the image is. It then finds the file name which is the last element in the array which was set up above.  Then there is a Try loop which checks that the file isn’t too big and is the correct file type once this was completed it then saves the image in the URL path which I set it as which in this case is the image folder in the website. It then sends a message that the message was successfully uploaded. However if the image file type is wrong then it outputs a message saying it’s the wrong file type or if the image is too large then it says that the file is over the limit.  At the very end of the code there is error messages incase anything goes wrong or if there is nothing in the browse section and they try and click upload then an error message will appear saying there is nothing to upload. |  |
| This shows the data grid view which it displays that there is nothing to display. The screenshot also shows the uploads button which once information has been looked for the user will click this. |  |
| This screenshot shows what the window looks like when the admin has clicked on the browse button, the file explorer opens and the admin can choose the picture which is required for that property. |  |
| This shows once the image has been selected and the admin as clicked open then the URL is displayed in the search bar next to the Browse button. The next step is it click on Upload, this then runs the code which is shown above. |  |
| Once the code has run this message will appear which will either display a warning message saying there was an error or if its run successfully then it will display the text as shown on the right. |  |
| This screenshot shows where the images will be inserted when the user clicked upload. All the images for the webpage will be saved here due to the code which is shown above. |  |
| This is what the property webpage will look like with everything been inputted onto one of the website pages. This shows both the text from the database and the input ways to insert an image. |  |
| The next stage was that to add this query, this is an insert query which will insert a value into the Picture table and insert the information into the column PictureURL. |  |
| I then inserted a table adapter at the top of the page called picupload and this linked to the query I just created above. Once I had done that I created a variable called saveimage. The saveimage variable is then used to store the image from the file upload. The code then goes onto using the picupload and the query created with the saveimage to insert the url which is stored in the save image to the database. This all came under once the upload button had been pressed |  |
| To show that it works I did I test. I used the upload button and I found an image. I then clicked the upload button and the file successfully uploaded to the database as shown. The next stage for the images to then to change the 0 to which ever property the image is associated to so either a 3 or a 4 then the image will be displayed on the website. |  |
| This is how I inserted the images into the webpage itself. It’s different to what is described above as that doesn’t actually insert the images into the webpage it only stores them. This shows what the code will look like to insert the images into the website. It uses the array created in the DGV to get the url of the image path once the path as been inserted into the database. |  |
| Once the code has run then the images are created and inserted into the website, however they at the minute are being inserted into the wrong section of the page.  To see how I changed this please see the Errors in development section where everything will be explained. |  |
| The next stage of developing the website is the Calendar Page. To begin this, I created an SQL select statement. The screenshot on the right shows that I selected the current renter from the renter table. Outputting the current renter information however I also selected ViewerID as this is what is used to select the right person from the table, that is why it (ViewerID = ? ) this is because it wants to know what the ViewerID is. I then saved this query using a suitable name so that I will know what the query does. |  |
| This is the SQL statement needed for the calendar page to work correctly. This statement outputs the date for payments from the renter table using the ViewerID to know which row of information to associate with. I then saved this SQL statement using a sensible name to tell me what the query does. |  |
| This code shows that a session is passed from the login page called ‘RenterID’ which was shown earlier in development. The ‘RenterID’ is passed into the calendar page. It then uses a query which we create above called ‘SelectCurrentRenterFromRenterID’ and inputs the RenterID which was passed to see if they are a current renter, if they are a current renter who have logged in then both the label called lbl\_prop1rent becomes visible so the use can see and then it also makes the label called lbl\_propertyrent1 show using another query which I created above called ‘selectDateOfPaymentsFromViewerID’ and passed the RenterID into this to then display the correct date for which the payment is due. It gets this information from the database which the admin needs to enter for this to be displayed. However, if a use tries to click on the calendar page without logging in then they will get a message box pop put saying they aren’t a current renter and get redirected to the login page. Also if a user tries to login and they aren’t a current renter and they try access this page then the same message box appears and are then redirected back to the login page to carry on looking around the website. |  |
| This screenshot shows the label which I inputted into the webpage from the tool box and changed the properties to remove the text and change the ID to lbl\_propertyrent1. This is so the information can be inserted from the database into the label as described above. |  |
| This shows the testing of the create user page. As you can see I have entered all of the details in the correct box and they all match. I then click the ‘create user’ button and then the page refreshes and the information which I inputted have now been cleared because of the clear code which I put on this page earlier. |  |
| This screenshot shows the information which I have inputted into the create user form as show above has successful been place into the database all in the correct order. Showing that the query and the form all work correctly as planned. Whenever a user enters information and submits the form to the database it automatically makes sure that admin = false so won’t tick the admin box. |  |
| The next stage of testing was to check that the login section of the login page works correctly. To do this I entered the email address and password which I had just entered into the create user form and I knew it had been entered into database table. Once I had entered the information as shown to the left by clicking the login button, I got the message below |  |
| The message box which I first received was the welcome message which I had created to welcome the user which shows that the query to the database works otherwise a message box with invalid email or password would appear. |  |
| However, the second message box which I received after the one above was the one which is shown on the right. This shows that the query I set up to make sure that they are a valid user before they get sent to the calendar page works, because this works it means that no unauthorised user will be able to gain access to the calendar page. |  |
| As shown above if they aren’t a current renter then they get a message. This next stage will check to see if the database table works along with the sessions and the queries which were set up. The screenshot on the right shows the database table ‘Renter’ which the admin must enter the ViewerID which can be found on the Viewerlogin table. By entering in the ViewerID it means that my client is able to select who has access to the calendar and select the current renters. Since the current renter has been selected and the date for payments has been filled in I can try logging in again. |  |
| As you can see on the right I have used the same email and password as used before. This time when I login I get the welcome message and no message saying I am not a current renter. |  |
| This page shows that the above has worked and I have been directed to the calendar page. Therefore, it also shows that the session works as its passed the ViewerID from the login page to the calendar page which then the queries also work as its selected the correct information from the database and outputted the date which the rent is due. Therefore I am happy that the testing of the login and calendar page does what I want and intended. |  |
| The next stage of development was to add better validation to the create user section of the login page. This is because the way which the validation check runs currently on the login page is it just checks that the emails match and the passwords match but don’t include any ways to check that the correct symbols are included in the email address as I could have just typed ‘email’ as my email address and this would have run whereas an actual email address should have the @ symbol. Because of this I decided to change all of the coded validation with the annoying message boxes and change it to more efficient way of validation.  To the right shows required field validation which will go on every textbox in the create user form as each piece of information is needed. I began by dragging across a required field validation from the toolbox and then I clicked on properties and edited the control to validate to whatever text box the validation was next to so in this case it was txt\_email, I then had to enter the CSSClass which was ErrorMessage as in the CSS document I had a section for this I also had to change the Display to dynamic so that it would change, the final stage for this type of validation was to include an error message to get displayed to tell the user what they didn’t do. |  |
| The second type of validation which I had to use was a regularexpressionvalidator what this is make sure that the information which is inputted for the email included the correct symbols for it to be a correct email address. This validation goes next to the email and confirm email address. I had to change the same things as described above but also had to change validationexpression to online email, this then shows all of the symbols shown in the screenshot to the right. |  |
| The final piece of validation which needs to be included goes next to the password and confirm password address. What this type of validation does it compares the two to make sure that they are the same and stop any error which may occur. The same settings are edited as described for the required field validator however for this validation we need to change the properties for the ControltoCompare to txt\_createpass and the ControltoValidate to txt\_createconfirmpass. This is used to compare the two and compare that they are the same. |  |
| Once the above is complete the create user form should like the screenshot on the right. I also dragged across from the toolbox a Validation summary which is the box at the bottom with Error message and 2. This is used to display all of the error messages if there are any. |  |
| I then had to change the code on the right so that the validation would work and information could be inputted into the database. To do this I commented out the function checkmatch because it was no longer needed as the validation was already taking place so nothing needed checking again like the email and password. I then could comment out the top line as this would prevent information getting inserted into the database until the checkmatch function had been completed. I had to keep the second line in order for the information to be entered into the database. However, the information was only inserted into the database once the validation checks created above are complete.  I did experience an error before I got to this which you can see in the section called Errors in development. |  |
| As you can see in this screenshot that the information is entered correctly with all the text boxes filled in and the correct symbols for the email address and the two passwords match. This means that this set of information will pass through the validation checks and be entered into the database. |  |
| As you can see the information has been successful inserted into the database in the correct order and exactly how the information was in the create user form. Therefore, I am happy with how the create user form works and that it includes everything which I want it to. |  |

## Errors in development

|  |  |
| --- | --- |
| This error on the right shows the code from the create user section of the login page. The reason why this code doesn’t work it due to the new validation which I used instead of the message boxes pop up. Therefore, I commented out the whole of the code which is involved in the message box validation, this meant that when I ran the code and tried entering information into the text boxes for create user when I clicked the create new user button the code runs however when I then checked the database none of the information gets inserted into the database. |  |
| To fit the error above I uncommented the top lines of code and then read through the code, I then realised that the top line of code is associated with the function checkmatch which I had already commented out when I changed the message box validation to the correct type of validation. This meant that the top line of code was not needed, therefore I could comment this out and just keep the second line. This meant that the information could be correctly be inserted into the database as shown by the screenshot, which I inserted information into the create user form on the webpage and clicked the submit button which the information then was successfully inserted into the Viewerlogin table. Therefore, the error has now been fixed. |  |
| The error which I experienced was when running the website in the internet browser, I clicked the webpage about from the navigation menu, then I got the error which is shown on the right. This error appears as when I created class base page and since each page inherits this class which means that the code will automatically if the page doesn’t have a title will produce the error page as shown on the screenshot on the right. When I then go to the source of the about page I noticed that I didn’t have a page title which is why the error appeared. |  |
| To fix this problem I added some text in the page title section of the code as shown by the screenshot on the right, once I had added and title and rerun the page, there was no error messages which appeared which shows that this was the only error involved in this page therefore I am happy that this error has been fixed. |  |
| This error is that whenever the user enters information into the create user section of the login page and then submits the information, all the information is inserted correctly but it also puts a tick on the admin tick box which isn’t correct as every user that creates an account would find that they are admin rather than just a regular user. This is due to in the second line of code at the end its set to true. |  |
| To fix the problem to prevent the admin box to be ticked in the Viewerlogin table once information from the create user form has been submitted, I had to change the code as shown on the screenshot on the right so that at the end of the statement it says false instead of true. Once I had done this I then tested to check that changing the code from true to false would work. The test shows that all the information entered is all in the correct place and also at the end the tick box isn’t ticked therefore the error has been corrected and everything works correctly as it should. |  |
| This is the error which I encountered when running the properties webpage as when in the database there was a url and the query was working and calling the correct information however no images where being displayed. |  |
| To fix this I had to cut down the absolute file path as the code couldn’t find the whole of the file path therefore couldn’t get to the image. To fix this in the database I made the file path relative and it runs from the file path from within the webpage folder itself. Therefore, cutting down the path in the database meant that the code could find the image as shown on the right and the images are displayed. However, they aren’t in the correct place, this is further explained below. |  |
| The error which I experience when running the website was the pictures weren’t in the correct place. To find out why, I ran the website in the browser right clicked and clicked on view source, I then went to the bottom of the debugger and found where the images are being displayed and from this I knew that there was something wrong as the code was in the total wrong page. |  |
| To combat this problem, I dragged 4 image boxes from the toolbar to where in the table I wanted them. I then went to the code and created an array which is seem as imagestring(3) which will mean 4 image urls can then be used. I then did imagestring(i) = url which will add the url into the array. I then used the next section of code to create a new image. Then from this I could get the url from the image created.  The block of code creates width and height of the image.  I comment out the me.controls like of code because I no longer needed this as this piece of code displays the images at the bottom of the page which I do not want.  The final block of code after the next pulls the url from the array and displays the images in the image boxes from the toolbox.  Then I ran the code to check that it worked correctly and as you can see on the right it gets displayed perfectly. Therefore, the image problem as successfully been fixed. |  |

## Reviewed and changed Algorithms

This changed algorithm is for adding images to the property website which was overlooked in the design section of the document. The below will show how I will add images to the property webpage.

1. If renterID = admin account then but\_uploadpic . visible = true
2. Grid view visible = true
3. File upload pic button visible= true
4. End if
5. For i as interger = 0 to grid view. Rows. Count – 1
6. arrayFromDVG(i) = Gridview.rows(i).cells(0).text
7. next
8. For i as decimal = 0 to array size
9. Imagestring(i) = url
10. Pic add( add new image)
11. Pic(i). ImageURL = url
12. Pic(i) visible = true
13. Pic(i) width = 100
14. Pic(i) height = 100
15. Next
16. Image1.Imageurl = imagestring(0)
17. Image2.Imageurl = imagestring(1)
18. Image3.Imageurl = imagestring(2)
19. Image3.Imageurl = imagestring(3)
20. Button upload clicked
21. If ( File upload pic is nothing) and (fileupload pic content length > 0)
22. Then
23. Get file name (FileUpload pic. PostedFile.Filename
24. Filetype = fileupload pic .PostedFile.Filename
25. FileType = Right(filetype, 3)
26. Try
27. If fileupload pic.PosedFile.ContentLength <=50000000 and filetype = ‘jpg’, ‘png’, ‘gif’ then
28. FileUpload pic. Postedfile.saveas ( wherever images are saved to &sFileName)
29. Saveimage = (end of address of folder & sFilename)
30. Picupload.InsertURLintoDB(Saveimage)

## Prototyping

|  |  |
| --- | --- |
|  | The way which I did prototyping is through using word to create a rough idea of how things will look with not adding a lot of detail.  For example, on the right shows the first prototype which I created for the login page. It doesn’t include a lot of detail it just states where information will go. I then used this to show my client to see what he thought of this page. He said he approved of the layout of the page, I then had a basic idea of layout of the page so when I created and designed it in VB I had something to go off.  As shown by the screenshot it shows that I pretty much went with the basic design. When I showed the client, he approved of this page therefore I didn’t need to do anything to change this. |
|  | The next prototype was the booking page. My client liked the idea of this page however I didn’t have enough time to create this therefore there is no page on the website for bookings. Instead of doing this I told my client that the users could organise visits through the contact us form.  This is the only prototype which I created as I didn’t need to change any information because I wasn’t going to use it. |
|  | This is the next prototype which I created. It shows the basic outline of the calendar page which users can use if they login and are current renters. The renter signed off on this page which meant that I could design the actual page around this.  The screenshot below is the final prototype which I created it shows the calendar page. It has the same design as the initial prototype, with the rent is due being in the same place and same with the information for renters.  Once I had created this the client signed off on this, meaning that I didn’t need to change anything or make any more prototypes, therefore this met the design brief and is a successful page so I am happy with this. |
|  | This shows the first prototype for the property page, it only shows the property master page and how I would layout this page. However, I didn’t create an initial page prototype but in VB I did create this page because I thought it would be best way to display the properties for users to then click on to choose the property.  This screenshot below shows the initial property page which displays just the address and price, the user can then click on the button underneath each property. This would redirect to the property master page. The client signed off on this page and liked how this worked meaning that I met his brief and in my view this is another successful page.  The screenshot underneath shows the actual property master page which I created following the design which my client liked. This meant that designing the look of this page was pretty simple to follow it was just the code behind this page was much more difficult. Even after this the page looks how it should and I am happy with how it appears. When I showed my client, he liked this page and was happy with how it works therefore this is another successful page. |
|  | This is the final prototype which I created, this was for the contact us page. This page layout is simplistic and would be easy for users to be able to fill out. When I showed the client this page he was pleased with how it would look. He signed off on this design and wanted to see the product after the final prototype/ final product had been created.  I then created the actual page for the contact us, the design follows the layout created in the first prototype. It looks effective and is in a systematic order which makes sense. When I showed the client this page they liked it and said that it was perfect. From this I would say it was a successful prototype and nothing needed to be changed. |

## Annotated modular code

CSS File

\*

{

/\*

Defines the main font used throughout the entire site.

\*/

font-family: Arial, Sans-Serif;

}

body

{

/\*

Clears white space around the body and gives it a grey background color.

\*/

margin: 0;

background-color: #cccccc;

}

a

{

/\*

Clears the underline on normal links, and gives them a different text color.

\*/

text-decoration: none;

color: #0760B2;

}

a:hover

{

/\*

Underlines links when the user hovers the mouse over them.

\*/

text-decoration: underline;

}

h1

{

/\*

Makes h1 elements smaller than their browser default.

\*/

font-size: 20px;

}

h2

{

/\*

Gives h2 elements a different color and size,

and adds a bit of margin at the top to create room between the heading and the preceding element.

\*/

font-size: 14px;

color: #3e3e3e;

margin-top: 10px;

}

#PageWrapper

{

/\*

Sets the total width for the entire page. Also sets the background color

which is used to fill the background of the Sidebar in case the MainContent section is taller than the Sidebar.

Uses margin: auto to center the entire page in the middle of the browser's window.

\*/

width: 844px;

background-color: #5487c0;

margin: auto;

}

header

{

/\*

Gives the header the same width as the PageWrapper. The height creates some room for the logo

that is set with the background-image.

\*/

background-image: url(Images/Header.jpg);

width: 844px;

height: 120px;

}

header a

{

/\*

The header a is a link nested in header. It provides a link back to the homepage.

The size of the link is the same as the header, so the entire header is clickable.

\*/

width: 844px;

height: 86px;

display: block;

}

nav

{

/\*

The menu spans the page width, right below the header.

At the top and left a few pixels padding is applied to create some room.

\*/

background-image: url(Images/MenuBackground.jpg);

width: 827px;

height: 36px;

padding-top: 7px;

padding-left: 17px;

}

nav a

{

/\*

Links in the Menu are white.

\*/

color: #fff;

}

.MainMenu

{

/\*

The Menu gets a white border on all four sides.

\*/

border: 1px solid #999999;

width: 814px;

height: 19px;

background-color: #555555;

}

.MainMenu ul li

{

/\*

Gives the 6 menu items in the main menu a width of 120px each.

\*/

width: 130px;

}

section#MainContent

{

/\*

Defines the main content area. The #MainContent element has a minimum height of 500 pixels, but can grow if necessary.

The font-size is 80% of its parent element, which in this case comes down to 80% of the font

the user has specified as the default font in the browser.

\*/

font-size: 0.8em;

width: 659px;

border-left: 1px solid white;

border-right: 2px solid white;

float: left;

background-color: #e1e1e1;

min-height: 500px;

padding: 10px;

}

aside#Sidebar

{

/\*

The Sidebar is positioned to the right of the MainContent section. It gets the same font-size as the

#MainContent section and gets a background image called Sidebar.jpg.

To ensure the image is visible in (most) browsers on a small page, the element gets a minimum

height of 500px.

\*/

font-size: 0.8em;

color: White;

background-image: url(Images/Sidebar.jpg);

background-repeat: no-repeat;

background-color: #7773A1;

width: 142px;

min-height: 500px;

padding-top: 10px;

padding-bottom: 10px;

padding-left: 20px;

float: left;

}

footer

{

/\*

The footer is positioned below all other content (yet still within PageWrapper).

clear: both is used to clear the impact of the float properties used for #MainContent and #Sidebar.

\*/

width: 844px;

clear: both;

height: 37px;

background-color: #808080;

color: White;

text-align: center;

font-size: 0.7em;

font-weight: bold;

line-height: 37px;

}

.Introduction

{

font-style: italic;

color: #3e3e3e;

}

.aspNetDisabled

{

color: grey;

}

/\*This styles the buttons on the webpages\*/

.MyButton

{

color: white;

}

ul.level1

{

/\*defines appearance of main menu items\*/

font-size:14px;

font-weight: bold;

height:19px;

line-height: 19px;

}

ul.level1 .selected

{

/\*defines apperance of active menu items\*/

background-color: #509EE7;

}

a.level1

{

/\*adds white space to left of main menu item text !important used to overrule the in-line css that the menu generates\*/

padding-left: 5px !important;

}

a.level2

{

/\*Defines the appearance of the sub menu items\*/

background-color: #555555;

padding-left: 8px;

}

a.level1:hover, a.level2:hover

{

/\*defines hover style for main menu and sub items\*/

background-color: #509EE7;

}

.ErrorMessage

{

/\*Error Messages for forms\*/

color:red;

}

.PleaseWait

{

/\*Adds the please wait gif\*/

height:32px;

width:500px;

background-image:url(Images/PleaseWait.gif);

background-repeat:no-repeat;

padding-left:40px;

line-height:32px;

}

.Attention

{

border:4px solid red;

padding: 10px 0;

width: 100px;

margin: auto;

display:block;

text-align:center;

}

Calendar Page

Partial Class About\_Calandar

Inherits BasePage 'inherits the class base page which I created.

Public currentrenter As New DataSet2TableAdapters.RenterTableAdapter 'create variable for the query created in the dataset

Protected Sub Page\_load(ByVal sender As Object, e As EventArgs) Handles Me.Load

Dim renterID As String = Session.Item("RenterID") 'passes session renterID from the login page

If currentrenter.SelectCurrentRenterFromRenterID(renterID) = True Then 'uses the query to input the renterID passed from the session to check if they are a current renter

'If its true then it makes the labels visible so the user can see what’s in the labels

lbl\_prop1rent.Visible = True

lbl\_propertyrent1.Text = currentrenter.selectDateOfPaymentsFromViewerID(renterID) 'in this label it uses the query in the dataset to select the payment from the viewerID

'it then uses the renterID from the session to then search for the person who has logged in to then input the data for payment from the database.

Else

MsgBox("You are not a current renter", MsgBoxStyle.Information, "Not a renter") 'If the person logged in isnt a current renter then a message box will appear telling them

'the message shown

Response.Redirect("../Login.aspx") 'once they have clicked okay on the message box it will redirect back to the login page.

End If

End Sub

End Class

Contact Form

Imports System.IO 'Provides access to the file class for reading the file

Imports System.Net.Mail 'Provides access to the various mail related classes

Partial Class Controls\_ContactForm

Inherits System.Web.UI.UserControl

Protected Sub CustomValidator1\_ServerValidate(source As Object, args As ServerValidateEventArgs) Handles CustomValidator1.ServerValidate

If Not String.IsNullOrEmpty(Mobile.Text) Then

args.IsValid = True

Else

args.IsValid = False

End If

End Sub

Partial Class Controls\_ContactForm

Inherits System.Web.UI.UserControl

End Class

Protected Sub SendButton\_Click(sender As Object, e As EventArgs) Handles SendButton.Click 'does the following when the send button is pressed

If Page.IsValid Then 'makes sure page is valid and validation is correct

Dim fileName As String = Server.MapPath("~/App\_Data/ContactForm.txt") 'gets text file from the folder

Dim mailBody As String = File.ReadAllText(fileName) 'reads text

'uses the varibale mail body which reads the text file and then replaces the mail body text file with whats in the text boxes. Makes the emails look identical

mailBody = mailBody.Replace("##Name##", txtName.Text) 'the information in the name text boxe takes over the ##name## so can be used in the email

mailBody = mailBody.Replace("##Email##", EmailAddress.Text) 'the information in the emailaddress text box takes over the ##email## so can be used in the email

mailBody = mailBody.Replace("##Mobile##", Mobile.Text) ' the information in the mobile text box takes over the ##mobile## so can be used in the email

mailBody = mailBody.Replace("##Comments##", Comments.Text) 'the information in the comments text box takes over the ##comments# so can be used in the email

Dim myMessage As MailMessage = New MailMessage() 'creates variable my message so new email messages can be created

myMessage.Subject = "Response from website" 'adds the subject to the email

myMessage.Body = mailBody 'adds the information stored in the mail body to the email

myMessage.From = New MailAddress("daisyleeholdings@gmail.com", "Sender Name") 'creates new emails from the email address shown

myMessage.To.Add(New MailAddress("daisyleeholdings@gmail.com", "Reciever Name"))

myMessage.ReplyToList.Add(New MailAddress(EmailAddress.Text)) 'when reply button pressed email address from textbox added

Dim mySmtpClient As SmtpClient = New SmtpClient()

mySmtpClient.Send(myMessage)

Message.Visible = True 'makes the message visible

MessageSentPara.Visible = True 'makes the label with the message visible once message is sent

FormTable.Visible = False 'makes the contact us form not visible once the message has been sent

End If

End Sub

End Class

Properties First Page

Public Class Properties\_Properties

Inherits BasePage

'makes the table adapter created in dataset visible in the other functions on this page

Public tableAdapter As New DataSet2TableAdapters.PropertyTableAdapter 'variable makes the propertytableadapter public

Public image As New DataSet2TableAdapters.PictureTableAdapter 'variable makes the picturetableadapter public

Protected Sub Page\_load(ByVal sender As Object, e As EventArgs) Handles Me.Load 'when the page loads the following is visible

Lbl\_prop1.Text = tableAdapter.Select\_Address(3) 'adds the query select\_address created in the dataset to the label called prop1.The query looks for the propertyID which for this is 3

Lbl\_prop1price.Text = tableAdapter.Select\_price(3) 'adds the query select\_price created in the dataset to the label prop1price. The query looks for the ProperyID which for this is 3

Lbl\_prop2.Text = tableAdapter.Select\_Address(4) 'adds the query select\_address created in the dataset to the label called prop2.The query looks for the propertyID which for this is 4

Lbl\_prop2price.Text = tableAdapter.Select\_price(4) 'adds the query select\_price created in the dataset to the label prop2price. The query looks for the ProperyID which for this is 4

End Sub

Protected Sub but\_propweb1\_Click(sender As Object, e As EventArgs) Handles but\_propweb1.Click 'when the button for property 1 is clicked it does the following

Session("PropertyID") = 3 'stores the propertyID which for this is 3. It stores the information as a session so can be passed between pages

Response.Redirect("../PropertyMasterPage.aspx") 'when the button is clicked it redirects the user the propertymasterpage

End Sub

Protected Sub but\_propweb2\_Click(sender As Object, e As EventArgs) Handles but\_propweb2.Click 'when the button for property 2 is clicked it does the following

Session("PropertyID") = 4 'stores the propertyID which for this is 4. It stores the information as a session so can be passed between pages

Response.Redirect("../PropertyMasterPage.aspx") 'when the button is clicked it redirects the user the propertymasterpage

End Sub

End Class

Properties master page

Public Class \_Property1

Inherits BasePage

Dim tableAdapter As New DataSet2TableAdapters.PropertyTableAdapter 'stores information propertytableadapter as a variable called tableadapter

Dim list As List(Of Integer) = New List(Of Integer) 'dynamic list to add extra elements

Dim arrayFromDGV() As String 'variable to make the array as a string

Dim tableAdapter2 As New DataSet2TableAdapters.PictureTableAdapter 'stores information picturetableadapter as a variable called tableadapter2

Dim picupload As New DataSet2TableAdapters.PictureTableAdapter 'stores information picturetable as a variable called picupload

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load 'the code below is run when the page loads.

Dim PropertyID As Integer = Session.Item("PropertyID") 'stores the information from propertyid which as been passed from the first property page as a variable called propertyid

Dim renterID As Integer = Session.Item("RenterID") 'stores the information from renter which as been passed from the login page as a variable called propertyid

lbl\_webpropadd.Text = tableAdapter.Select\_Address(PropertyID) 'gets the information using the query select address and uses the session passed to input the correct infromation for that property from the database

lbl\_Bathroom.Text = tableAdapter.SelectBathroomNumber(PropertyID) 'gets the information using the query selectbathroomnumber and uses the session passed to input the correct infromation for that property from the database

lbl\_Bedroom.Text = tableAdapter.Select\_BedroomNumber(PropertyID) 'gets the information using the query selectbedroomnumber and uses the session passed to input the correct infromation for that property from the database

lbl\_Description.Text = tableAdapter.Select\_Description(PropertyID) 'gets the information using the query selectdescription and uses the session passed to input the correct infromation for that property from the database

lbl\_price.Text = tableAdapter.Select\_price(PropertyID) 'gets the information using the query selectprice and uses the session passed to input the correct infromation for that property from the database

lbl\_type.Text = tableAdapter.Select\_prop\_type(PropertyID) 'gets the information using the query selectproptype and uses the session passed to input the correct infromation for that property from the database

lbl\_Description1.Text = tableAdapter.SelectDescriptionTwo(PropertyID) 'gets the information using the query selectdescriptiontwo and uses the session passed to input the correct infromation for that property from the database

lbl\_Description2.Text = tableAdapter.SelectDescriptionThree(PropertyID) 'gets the information using the query selectdescriptionthree and uses the session passed to input the correct infromation for that property from the database

If renterID = 2 Then 'if the renterid = 2 'this is becasue renterid 2 is the admin account

but\_uploadpic.Visible = True 'if =2 then it makes the upload button visible

GridView1.Visible = True 'makes the gridview visible

FileUpload\_pic.Visible = True 'makes the fileupload pic visible

End If

'Sets arraysize = number of rows

Dim arraySize As Integer = GridView1.Rows.Count - 1 'counts number of rows in DGV and - 1 to redim array

ReDim arrayFromDGV(arraySize) 'redim from DGV to set size of array

'adds each part from the grid to the list

For i As Integer = 0 To GridView1.Rows.Count - 1

arrayFromDGV(i) = (GridView1.Rows(i).Cells(0).Text) 'each row it adds text to populate the array

Next

Dim Pics As New List(Of Image) 'sets up picture element

Dim imageString(3) As String 'creates array for the images can only have a max of 4 images can be used

For i As Decimal = 0 To arraySize 'gets url for every image

Dim url As String = arrayFromDGV(i) 'starts the array to get the information from the DGV table.

imageString(i) = url 'adds the url into the array

Pics.Add(New Image) 'creares a new image

Pics(i).ImageUrl = url 'gets the url for the image created

Pics(i).Visible = True 'makes the image visible

Pics(i).Width = 100 'sets the width of the image

Pics(i).Height = 100 'sets the height of the image

'Me.Controls.Add(Pics(i)) 'adds the image 'didnt need this due to the array created to display the images

Next

Image1.ImageUrl = imageString(0) 'adds the information saved in the array to the Image picture dragged from the toolbox

Image2.ImageUrl = imageString(1) 'each one from the array will be displayed in this.

Image3.ImageUrl = imageString(2)

Image4.ImageUrl = imageString(3)

End Sub

Protected Sub btn\_upload\_Click(sender As Object, e As EventArgs) Handles but\_uploadpic.Click

'check that the file has been selected and it's a valid file

If (Not FileUpload\_pic.PostedFile Is Nothing) \_

And (FileUpload\_pic.PostedFile.ContentLength > 0) Then

'determine file name

System.IO.Path.GetFileName(FileUpload\_pic.PostedFile.FileName)

'get the full path of the image

Dim pather As String = FileUpload\_pic.PostedFile.FileName

'use split and delimit by \ to break up path

Dim pathSplit() As String = Split(pather, "\")

'find file name, which is the last element in the array

Dim pathSplitSize As Integer = pathSplit.Length - 1

'set the uploaded file name to the path file name.

Dim sFileName As String = pathSplit(pathSplitSize)

Dim fileType As String

'find the file type

fileType = FileUpload\_pic.PostedFile.FileName

fileType = Right(fileType, 3)

Try

Dim saveimage As String

If FileUpload\_pic.PostedFile.ContentLength <= 500000000 And fileType = "jpg" Or (FileUpload\_pic.PostedFile.ContentLength <= 500000000 And fileType = "png") Or (FileUpload\_pic.PostedFile.ContentLength <= 500000000 And fileType = "gif") Then

'save file on disk

FileUpload\_pic.PostedFile.SaveAs("C:\Users\William\Documents\Visual Studio 2015\WebSites\WebSite2\Properties\Pictures\" & sFileName)

saveimage = ("\Properties\Pictures\" & sFileName) 'stores the file uploaded

picupload.InsertURLintoDB(saveimage) 'inserts the save image stored into the database

lbl\_Message.Visible = True

lbl\_Message.Text = "File: " + "C:\Users\William\Documents\Visual Studio 2015\WebSites\WebSite2\Properties\Pictures\" & sFileName +

" Uploaded Successfully"

Else 'reject file

If fileType <> "jpg" Or fileType <> "png" Or fileType <> "gif" Then

lbl\_Message.Visible = True

lbl\_Message.Text = "Image file must be jpg, gif or png."

Else

lbl\_Message.Visible = True

lbl\_Message.Text = "File Size if Over the Limit of " & 5000000000

End If

End If

Catch exc As Exception 'in case of an error

lbl\_Message.Visible = True

lbl\_Message.Text = "An Error Occured. Please Try Again!"

End Try

Else

lbl\_Message.Visible = True

lbl\_Message.Text = "Nothing to upload. Please Try Again!"

End If

End Sub

End Class

Login Page

Partial Class Scripts\_Login

Inherits BasePage

Public loginViewer As New DataSet2TableAdapters.ViewerloginTableAdapter 'make the dataset readable by the code below by calling it login viewer

Protected Sub But\_createuser\_Click(sender As Object, e As EventArgs) Handles But\_createuser.Click

If Page.IsValid Then

'If checkmatch(txt\_email.Text, txt\_confirmemail.Text, "Emails don't match") And checkmatch(txt\_createpass.Text, txt\_createconfirmpass.Text, "Passwords don't match") Then

loginViewer.InsertNewViewerLogin(txt\_forename.Text, txt\_surname.Text, txt\_dob.Text, txt\_email.Text, txt\_createpass.Text, False)

'clears contents from each textbox once button is pressed

txt\_forename.Text = ""

txt\_surname.Text = ""

txt\_createpass.Text = ""

txt\_createconfirmpass.Text = ""

txt\_dob.Text = ""

txt\_email.Text = ""

txt\_confirmemail.Text = ""

End If

End Sub

'Function checkmatch(ByVal check1 As String, ByVal check2 As String, message As String) As Boolean

' If check1 = check2 Then

' Return True

' Else

' MsgBox(message, MsgBoxStyle.OkOnly)

' Return False

' End If

'End Function

Private Sub But\_login\_Click(sender As Object, e As EventArgs) Handles But\_login.Click 'when login button is pressed then does the following

Dim r = Me.loginViewer.LoginSelect(Me.txt\_loginemail.Text, Me.txt\_password.Text) 'uses the variable r to store the email and password which will then be used to input

'into the login select query to check if the email and password is valid.

If r Is Nothing Then

'checks if the txt\_loginemail Or txt\_password wrong

MsgBox("Invalid Email or Password", MsgBoxStyle.Information, "Invalid credentials")

Else

'Valid email & password

MsgBox("Welcome to Daisy Lee Holdings", MsgBoxStyle.Information, "Welcome") 'uses message box to display the welcome message if the info is valid

Session("RenterID") = loginViewer.SelectViewerIDFromEmailandPassword(Me.txt\_loginemail.Text, Me.txt\_password.Text) 'stores the infromation from the login page as the session called 'renterid'

'to pass information to other pages in the website.

Response.Redirect("about/calandar.aspx") 'redirects to the calandar page

End If

End Sub

End Class

## Testing to inform development and Evaluation

|  |  |
| --- | --- |
| To begin the testing, I started on the Login Page, with the create user section. As seen I didn’t enter any information at all I just clicked create user. This was checking that all the validation checks work correctly. |  |
| I then just entered text in the forename section and clicked create user. I did this to check that the validation worked correctly and didn’t submit any information into the database until everything else was filled in. This test worked successfully. |  |
| The next stage was to enter both the forename and the surname section. Once filled in I then clicked create user let the validation checks take place and as seen on the right there are still validation checks taking place therefore nothing will be entered into the database. Shows another successful test. |  |
| The next stage was to entered the above information and the date of birth as shown on the right, I then clicked create user and showed another successful test as nothing was entered in the database table. |  |
| The next stage was to enter an email address which didn’t include the correct characters which an email needs therefore when I clicked submit the following validation error appeared saying that I needed to enter a valid email address along with a password. Therefore it shows a successful test to check that it is a valid email with the correct symbols needed. | ` |
| I then entered the same information as before but this time in the confirm email address I entered confirmemail which neither matched the first email address or included any of the correct symbols required. From this I then got validation error messages saying to enter a valid email address and that the emails don’t match. This shows another successful test. |  |
| The next stage was to enter email addresses which match and contain the correct symbols. From this I, then clicked create user and nothing went into the database due to the form still not being complete as there if still the password section to enter. |  |
| I then entered a password as shown and clicked on the create user button therefore showing that the validation on the confirm password works as the error message appears. |  |
| After I entered in the confirm password which I just kept to pass as I wanted to check the validation between the password and the confirm password. The result found that the validation works and doesn’t submit the form to the database table as they don’t match, showing this test as a success. |  |
| I then changed the password to match and clicked create user. This tested that the query works inputting all of the data into the database table along with the button works. As shown on the right I found that the data was successfully entered into the table showing that the query works in the way I intended. |  |
| The next stage of testing was still on the login page but this time it was looking at the Login section. The first test I did was to try input the wrong email. I then clicked the login button and found that a message box appeared informing me that there was an invalid email or password. This message box shows that the validation process was successful and shows that the query works which calls the information from the table and uses the information to cycle through and compare the information which the user has inputted and the information which is in the table. Therefore, this test was successful. |  |
| The next stage was to test the login system using the correct email and password which was created in the testing above. Once I had inputted this data which I knew was correct I clicked the login button, the query successful ran and checked the information entered and that which is in the database table. Because both the information matched it meant that the message box appeared telling the user that they were successfully logged in. |  |
| However once the user had logged in the message box on the right appeared because they aren’t a renter which. This tested the query created which checks the renter table to see if the renter button is ticked corresponding with the correct Viewerlogin ID which was created when the user made an account. The only person who can change this is the administrator. This test therefore shows that this works correctly and displays the correct message when the user tries to access the calendar page. From this I am happy to say that this test was successful. |  |
| The next stage of testing was to change the renter to a tick for the user so I could test that they could access the page. Once I had ticked the renter box I could then access the calendar page which showed that this was a successful test. |  |
| The screenshot on the right shows a test of the calendar page, to test this I changed the date in database table, as shown. This was to test the query works correctly and as shown on the right the page updates with the correct information. I am happy that this test was successful and the components involved worked as expected. |  |
| The next test was on the property page; this is where the user would go to when they click on properties on the navigation menu. To test this I changed the information in the database table properties so that the header of each column was in each row under that column. For example, the text read address under the address column etc. This was to test that the query was getting the correct information along with it being displayed as I wanted. From the screenshot on the right I can conclude that for this page that the information is in the correct place and the queries works. |  |
| This is the page which the user can go to once they have clicked on the button ‘Click to view property’. I can conclude that the buttons works correctly and sends me to the correct pages when presses and the correct information is sent to the page depending on which property it is associated with. This means that the queries and sessions work properly so the test is successful. |  |
| These screenshots show the text once I had finished testing. I inputted data about two of the properties in the database, refreshed the pages and the information updated to what I wanted it to show. This shows that the testing was successful as I knew that the information was in the correct place. |  |
| The next stage in the testing process was to test the images for each of the pages and to make sure that the queries and array works to collect the images and display them.  To test this I got images off the internet of ranging from number 1 to 8. This is because there are only two pages which each can have 4 images on each. I inputted the url of where each picture was stored which can be seen on the screenshot. Then for images 1 to 4 I gave them the propertyID of 3 and then from 5-8 I gave them the propertyID of 4. |  |
| I then ran the website and clicked on the first property and as you can see the images are clearly displayed in the correct order and are on the correct page. After I then ran the second property and the images are displayed in the correct order. From this I can say that the testing worked successfully and display correctly. |  |
| The final stage of testing for the property page is the test of the upload button. To test this I had to login as an admin which I created earlier. The admin account just displays the uploads button so that they can add images easier than them having to find the url and input it themselves. To test this I used a red and a green apple image. I found the images which were saved in my documents, I then clicked open on the pop up window and then I clicked the upload button. The two screenshots below show the url which the query and code uploaded to the correct table and the correct column showing that it was a successful test. Overall for the property page I am pleased that the testing was successful and everything worked as intended. |  |
| To test the contact us page I filled in the name text box with the text name. I then clicked send to check that the validation features work correctly. As shown on the right all the validation error appear as there is nothing in those text boxes. Therefore, from this test was successful. |  |
| The next test was to enter an email address. I began by entering an invalid email. This is because it doesn’t include any symbols which need to be included for the email to be valid. I then clicked send and the validation error message appeared at the bottom of the page as shown on the right. This shows another successful test. |  |
| The next stage of the validation was to test the confirm email address against the actual email address to test this I entered confirmemail which didn’t match the email address therefore would produce a error validation message which it did telling the user it didn’t match. Therefore, showing that this worked as it should. |  |
| Once I knew the above worked I knew I had to change the emails so that they contained the correct symbols which an email must contain and I knew that I also had to change so that they matched. Therefore I used [email@email.com](mailto:email@email.com) and then clicked send. Once I have pressed send the validation messages appeared saying that I needed to enter a mobile number and to enter a comment. This meant that no information could be sent to the email address of the admin and also shows that this test was successful. |  |
| The next stage of testing included that date of birth section although this wasn’t a required field due to the information no being overly important at this stage. I still inputted to make sure that it worked in regards to text being entered into the text box and it was included in the email sent to the admin, which is my client. I then clicked send and error validation messages appeared. This shows another successful test. |  |
| This is the final stages of the test for the contact pages. For this I tested the mobile validation by entering in an 11 digit number and clicked send, this was successful and recognised that it was the correct length therefore only came back with the enter a comment validation message.  I then entered ‘test’ into the comments section and clicked send. Once I had done this the text boxes went because all of the validation was successful and the message which I created appeared. This shows a very successful test of the contact page with all validation and functions working correctly. |  |
| The next stage of testing was to make sure that the code which I created connects to the email server and displays the message which should have been sent of the contact page using the text file as the standard message format.  The first stage was logging into the account which I did successfully and then I clicked on the email which had just come through from the website.  The code linked to the text file making it a standardised message which inputted the information in the correct place. I am really happy with how this works and that it was a very successful test. |  |
| This was the final piece of testing which I needed to conduct on the page. What I was testing was that the code which I wrote worked. And as shown on the right the code successfully worked as the email address was added to the reply section of the email therefore makes it easier for my client to reply to the query or comment somebody makes. |  |

## Reviews

Once I had completed the above sections I then went to my client and showed him what I had developed. The client was very pleased with what I had developed and thought that the colour scheme looked smart professional.

I then took the client through each of the pages of the website showing what was on each page and if there was anything that he wanted to change. I began be showing the navigation menu and was pleased with how it functioned and liked how there was information about each page when the user hovered over each section.

The next place I showed him was the Home page however this doesn’t have any information on due to this not being one of the required fields which I was informed of but was there was the client to fill in with the relevant background information about the business if he chooses to do so. This was the same with the FAQ, About and with the review pages of the website.

Then I began to show him the pages which I developed. The first as in order of the navigation menu was the calendar system. I showed the client that if he didn’t have an account and wasn’t changed in the database then he wouldn’t have access to the page which he liked because at first I thought it might not be possible. I then had to divert to the Login page as I had to create an account to gain access. To do this I showed the client how all the validation worked, which he was impressed I then had once I had created an account had to show how it linked to the database and where the information was. Once in the database I showed him how to change the information so it allowed the user I just made have access to the page and display the correct date to which the rent was due. Once I had shown this I then went back to the login page and showed how the validation worked on the actual login part and then I entered the information which then went to the calendar web page and showed the date which I had just added. From this the client was extremely happy and liked how it all worked.

The next page was the properties page. I first began by showing the first page allowing the client to click the button depending on the page which he wanted to look at. This then took him to the property master page which showed all the information regarding that property. He was happy with how it looked also he liked how many images were on display and that they were the correct size. I showed him the database for this information and how he would be able to add and change the images if needed. He liked how this could be changed depending on the property which was for rent.

From the meeting, overall he was extremely please with how it all was and the way which it operated. I asked if he had any changes and he couldn’t think of any therefore I would say this was a success.

# Evaluation of solution

In this section I will go through the Decompose the problem so say which I have completed and if any that I haven’t completed.

The first part of decompose the problem is the login page. I had to make sure that it was accessible from the navigation menu which it is due to having its own section of the navigation menu that users can click on and they are automatically diverted to this page. I then had to have a webform which the users can fill in and then this information is sent to the database, I successfully did this which then allowed my client to change who has access to certain pages. For example, he could change that a user has access to the calendar page because they are a current renter so when the user logs in they will either get access to the page or they won’t depending on what my client has set the user privileges as.

On this page my client also wanted validation checks so it checked if the email addresses matched along with having the correct symbols for it to be valid. I also had to put validation checks on the passwords to make sure that they were filled in and that they matched each other. I successfully did this but I also went a step further and put validation checks on the rest of the fields including forename and surname as I believed that this was important otherwise in the database you wouldn’t know which account was linked to which. So now users must include forename and surname along with email and password for the form to be successfully uploaded to the database. This was all successful and does all upload to the database when all required fields are filled in therefore I am happy that I met the brief for this page.

The next thing which was a required field was the navigation system the client wanted a standardised colour text and spacing, I successfully did this using CSS which gave the navigation menu a darker grey then the actual page background. This makes it stand out and easy for the user to see also it does match the grey theme of the website therefore matches another specification which the client wanted. Each of the headings in the navigation menu all have the same spacing which makes it look professional and is wanted for a property website.

Another specification which the client asked for was that each of the pages are standardised, this is done once again through the CSS file which you will able to see in the section Annotated Modular Code. The pages all use a master page which I created, this makes sure that each of the pages have the same colour, same size header and side bar etc. and uses the same text size. I successfully did all of this and met the brief given.

The next page required was the property page, my client wanted information to be in a logical order with images. He also wanted the information to be stored in a database so that the information could easily be edited. I successfully created this with all the information being stored in the database so the client can edit if needed also the information is displayed in a logical order because the address is the main thing which the user will see so that they know which property is which, also the other information like price and description is further down the page after the images. The images are also stored in the database so that extra can be added if needed and the images are then displayed in each webpage depending on the propertyID. Overall this meets the client’s specification of which he wanted on the website.

The Booking system which the client wanted so users could choose a date and time to view a property unfortunately couldn’t be met this is because I ran out of time for it to be accomplished, this could have been helped if I had more time. The client when told understood why this wasn’t completed but was happy that it could be organised through the contact us page. However instead I created a calendar page which the users who were current renters could access if logged in so that they could see when their rent is due and the client liked this idea so he could edit the information in the database each month or if there was a new renter so they knew when it had to be paid. In a way this was a compromise for the booking system.

The final page which my client wanted in his brief was a general enquires form which I have called in the website contact us to make it sound more user friendly. The client wanted information like name email mobile number data of birth and their message collected from users to get a full picture of them. He also wanted validation checks to take place to make sure that the information has all been filled in and that the email addresses match. The final thing which the user wanted was the information to connect to his email address so he can access this anywhere on the go. I successfully completed this with all the validation checks being completed and the information gets sent to the client’s email, I also went a step further and made it so that all the information is presented using a standardised way with the information being presented in a logical way. The client is very pleased with how this works.

Overall the client was extremely happy with how the website looks and the actions which it performs. He also liked how it met his specification which he set and wasn’t particularly bothered about the booking system as the replacement with the calendar system was more than he expected and the contact us works better than the current system he has in place.

## Testing

Through testing I tried two things the first was to check that all the functions which I created worked and the second was to try break the program in any way to test robustness. From the testing, which I did I found that all my functions worked correctly as they should, this was a positive for my project because it meant that the website worked how it should with things getting entered into the database as they should. With information being entered into the database correctly it means that the login page calls the correct information allowing certain users to successfully login, it also means that the information for the property page works getting the correct information in the right order meaning that these pages all work really well.

Through testing I found no error which I hadn’t already corrected in development. Therefore, testing was successful. If another developer decided to add more to the website, they will need to look at my testing plan which can be found in testing to inform development this will mean the new developer will understand how I tested things in order to keep the high standard of functionality of the website.

## Usability features

This will evaluate the usability features which I created in the design section. The first feature which I included in the report was the text which I said for it to be successful had to have same size text and the same font for the main text. I successfully completed this using the CSS, this then meant that all size of the font was the same along with the colour of the text was black to then stand out from the background. This when asked users to look at the website they said that the black text helped them to see the information standing out from the grey background.

Another usability feature which I thought needed to be included was the buttons on each page being a blue colour with white text to contrast the background colour and making it stand out from the page so users would know that they need to presses the buttons in order for processes to happen. I successfully did this, when I asked a user of what they thought of this and if they believed it would help, they agreed saying that it’s a useful feature as it meant that people know what to do rather than it being the same colour as the background and people getting lost not knowing what to press.

The next usability feature which I mentioned was the use of message boxes, the client wanted these to inform users of what might have gone wrong etc. An example of message box which I included in the websites as on the login page when the client successfully uses the correct email and password then they will get a message box appear telling users ‘welcome to daisy lee holdings.’ This is another successful usability feature because it met what the client wanted and allowed users to know what’s happening.

The penultimate usability feature which the client wanted was the use of validation. The validation was used to check that users entered information correctly and then if information is missing then it informs the user that this information is required. This was successfully tested and when information wasn’t entered then it gave errors with information which needed to be entered. The information given was sufficient for users to know what they needed to enter. I would say that this works successfully and does everything which the client wanted. I also asked a user to have a say on this usability feature and they found it usefully and mentioned that it would be even more useful for older generations who might not know what to enter therefore the information would be enough for them to enter information correctly.

The final usability feature which my client wanted included was the use of labels this was more for the client’s usability then for the users. The use of labels was useful because it meant information could be added to the labels through queries, information from the database could then be added to these labels meaning the client could easily change this information without having to know how the website works. This was successfully tested which can be found in testing to information development and evaluation. I didn’t find any problems with using labels apart from for the description only a certain amount of information can be entered therefore I had to include description 1,2 and 3 to include all the information required. However, this meant that information had to be split to around 39 words for each description in the database. The client was happy with this usability feature and so was I, therefore it passed the brief.

## Evaluation

The website which I created I found that it pretty much matches to the word what the client wanted out of the website. He wanted a colour scheme which showed class which it does as it’s a grey theme which does look classy in my eyes. He also wanted certain things in the navigation menu which it does. Along with things like a contact page, login page and a properties page which is all included along with successful validation stages which makes sure that users enter information correctly, therefore this meets the success criteria which my client set for this to be a successful project.

## Success requirement not met

However, my client did say that he wanted a booking system which meant users could request a time slot to have a view of the property which they are interested in. I didn’t have enough time to include this in my project for it to look professional enough for myself and my client to be satisfied with, instead I created a calendar system. The calendar system wasn’t included in the success criteria properly as described above he wanted a calendar booking system instead of this I thought it would be a good idea to have this page still be instead create it so it just displayed when the rent is due for those who are renting properties. This works by when the renter logs in my client in the database would have given them access to this page and then the session is passed from the login page to the calendar and this will then display a date which the client can change. My client when show this liked the idea and so it was included in the website. My client then went onto saying that because of the way I had designed and set up the contact page this could easily be used as the booking form.

The contact us page was very successful as all the validation worked well on this page, it also meant that once users had entered all the correct information then this would be sent to my client’s email address, this is way he believed it could be used for bookings as the important information like the name email and mobile number was on this form along with any messages which users could fill in to tell him which property they are interested in. Also, it was set up so when my client clicked on reply in the email it would will in the email the user filled in therefore making it quick and efficient, this email concept my client liked as it’s a massive improvement than the current system which he uses and means that he can work anywhere at any time and stops him from having constant phone calls.

## How it could be improved?

When thinking about this website there are a few things which could be added to improve this site even more. The first which I believe would improve the look is on the property first page having an image of the property so that there is something visual for users to look at. The second improvement which I would add would be on the property master page having it so you could have more than just 4 images included because this can limit what users see. Another improvement which I can think of is maybe including attachments maybe PDFs for things like gas safety certificate which some users might want to see before they even consider having a look around the property. The final improvement which I believe might be beneficial is that on the reviews page maybe having an automated comment system which the user can fill in and say how many stars and leave a comment but had to include the property address so that others can then filter through the reviews to see what people are saying about these properties. Also in the property reviews you could use a filter which users can then select which property they want to see reviews for and these then appear making it easier for people to be able to find information.

This website met all the required specification, usability features and the sign off on all the pages which the client and myself figured out therefore I am pleased that this can be handed over to the client with everything working to the best of my knowledge and any improvements and changes which he decides he wants he can ask for support or any information which is required to do this process.

## Maintenance

Maintenance will need to be used to make sure that the website is kept up to date along with making sure faults with the page are kept to a minimum. The way which this website will need to be maintained is thought adding and removing properties depending on which become full and not for rent anymore, also the properties page will need to be kept up to date with relevant information also new images of required. If the client decides that he wants to change any of this information all he needs to do is change this in the database. By changing information in the database, it means that the website will not need to be down as much for maintenance, as soon as information is updated in the database and saved the webpage when reloaded will have any new information.

In regards to maintenance of the images the client can either update manually be typing in the url of the image or the best was of adding images to make everything easier would be to use the picture upload feature which is found on the property pages. This means that all the client needs to do is add the propertyID numbers which is related to each picture.

Another piece of maintenance would be within the database itself the reason why the database would need constant maintenance is to keep those who are renting up to date with the correct date of payments through the use of text in the renter table. The client will need to keep check of this system, also he will need to make sure the ViewerID is copied across correctly and that the renter box is ticked to make sure that the users who are renters get access to the calendar page. The client will also need to take off people’s access to the calendar system if they end up moving out to prevent access to the calendar system.

Maintenance may also include adding new features which the client might want later in the website. For example, at some stage he might decide he wants to add a booking system, change headers or even change the colour scheme therefore I would have to create a basic manual of how I created the website maybe even use part of this, this would be needed as another developer might be used and because people go about solutions in different ways they might not know what I did so using an instruction manual would make it easier.

# Bibliography

*Microsoft Website*. (n.d.). Retrieved from https://msdn.microsoft.com/en-us/library/8z6watww(v=vs.110).aspx

Zoopla. (n.d.). *Zoopla Property Renting Website*. Retrieved from http://www.zoopla.co.uk