Final Year Project Report

Half Unit – Final Report

A study in Human-Computer Interaction(HCI)

Harveer Dole

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science & Mathematics

Supervisor: Emma Lieu

Department of Computer Science
Royal Holloway, University of London

January 10, 2021

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 14547 including appendix, 13098 without appendix

Student Name: Harveer Dole

Date of Submission: 11/01/2021

Signature: HS DOLE

Table of Contents

Contents

Abstract

I am doing this project in Human-Computer Interaction because I believe creating a user interface which is easy to use for people is important in software development, a field I am interested in for the future. Furthermore, this will be the first time I will be doing a project with no real specification and limited help so this will also teach me project management skills such as time management and organization. Therefore, by doing this project I believe I will be able to gain both valuable and new skills which can help me in the future as this field has good career opportunities. At the end of this project I expect to make three different interfaces. These three interfaces will be a website, a GUI and an android app. The website I will create will be to do with shopping as I believe this is a popular field so I would easily be able to draw inspiration for ideas online, more specifically it will be somewhere where a user can view clothes and add them to their basket. The GUI will be a database interface where a user can add information to the database and view the current data in the database. My GUI will be based on a database as I have previous experience in this field and I believe this can help me as I implement it into a interface. The app will be a restaurant service where a user can view a menu and order their food as this is something a lot of places are doing due to the pandemic and so it can be useful. Also, as a lot of restaurants have gone down this route recently due to the pandemic this will allow me to also draw inspirations for ideas. In addition to this, it is a useful interface to have as it reduces restaurant training and makes it easier to get the correct orders in. I expect these interfaces all to be fully working and have good designs implemented which follow as many Nielsen's useability heuristics as possible. By following these heuristics, it will demonstrate a good designed user interface which is my end goal.

This report will consist of ten chapters, starting with Project Aims & Motivations. Here I will be explaining what my aims are for the project and the motivations behind them. The next chapter will be the Literature Review, this consists of me critiquing my most used resources throughout the project. Then we will have the Human-Computer Interaction (HCI) chapter which will go through theory relating to the topic so what exactly is HCI and its importance, the issues that require solving and the main principles of it. The Software Engineering chapter is next, this will be me discussing all the software engineering techniques I have deployed in my project. After that will be the POC Development chapter, here I will discuss and evaluate my proof of concept programs that have been made. Following on from this chapter will be the Design Decisions On Final Programs, here I explain all the reasons behind my design for my three interfaces. My next chapter is How To Use The Programs, this will show you how to correctly use my interfaces I have created. Then we go onto the Evaluation Of Three Interfaces chapter where I evaluate my interfaces I have created against HCI principles and issues. The penultimate chapter is my Self Evaluation & Professional Issues, in the Self Evaluation part I discuss how discuss how I feel the project went, what I learnt etc while the Professional Issues covers the topic Licensing. My final chapter is the Conclusion, here I go in to how the project went, what I would do if I had more time and if I met my original aims. At the end we also have an appendix , this will include the project specification, the installation manual and a section discussing the directory format that has been deployed.

Chapter 1: **Project Aims & Motivations**

My main motivations behind doing a project in a study of HCI is because I believe HCI is mostly to do with solving problems and innovation. ('What is Human-Computer Interaction (HCI)', 2019). HCI stands for Human-Computer interaction and it studies the design of machines so that they can serve their users as best as possible. HCI focuses on creating a natural dialog between the users and the machine so that the user does not require a lot of cognitive effort ('A Guide to Human-Computer Interaction (HCI) | Adobe XD Ideas', no date).In simple terms, the goal of HCI is that when a user interacts with a machine they should be able to accomplish their intended task as quickly and easily as possible with not much effort going into figuring out how to. The field allows you to create better services and products than ones that have possibly already been made. Furthermore, HCI is a massive part of our present and future as a lot of people make use of it in their everyday lives, this means there is a good chance many career opportunities exist regarding this field. By carrying out this project it will then allow me to learn the required skills which future jobs may be interested in.

My original project aims are to make 3 different interfaces which are all made using different languages and platforms. This will mean I get experience in software's and languages which I have not used before. Therefore, I will be a more experienced software developer and designer which are qualities companies look for when hiring. On top of this by trying to implement the 3 interfaces with different languages and software's it gives me a chance to work independently and out of my comfort zone. This is again something I believe is required in the real world as if you get a job especially to do with software development you won't always know what to do for a project, it would require research and learning, similar to what I am doing for this project. This is why I am making an app using C#, a GUI using Java and a website using JavaScript, HTML and CSS.

I want the interfaces to implement a certain amount of functionality. This means I want them all to be useable for their purposes, for example I'm making an app for a restaurant so I want the user to be able to order their food, for my website I want the user to be able to view the products of the business and acquire/purchase them and for my GUI I want the user to be able to edit data in a database. By implementing functionality, I hope to further improve my coding skills on the different platforms I choose. This will help broaden my skills on other software's and language which will again make me a more experienced/skilled software developer. Furthermore, by implementing this functionality I can study interaction between the user and machine, enabling me to find out if it meets HCI goals as discussed above. I want all these interfaces to be easy to use and pleasing for users to use on a regular basis.

The interfaces should try to demonstrate good design principles which follow as many as Nielsen's Heuristic model as possible as this will demonstrate a well-designed User Interface. Nielsen's Heuristic model are principles for interaction design that help to identify problems associated with the design of user interfaces (Muniz, 2016). For example, one of the principles is to have a match between system and the real world so we can check our design to see if it speaks the users language and if not we can correct it (Nielsen, 1994). By developing interfaces which match Nielsen's Heuristic model, a highly known and respected model I believe I will be able to improve my software design skills, something which I have not done much of. This can help me in the future for jobs to do with software design and will also allow me to perhaps show case my project work to demonstrate I possess the correct skills.

Chapter 2: **Literature Review**

I found that one of the most-used literature's I used that I found useful when carrying out this study in HCI was Steve Krug's book titled Don't Make Me Think. I found this book very useful when designing my interfaces as it provided detailed explanations of what to avoid and what to include, followed by the explanations of why. This enabled me to easily understand what users would want from the systems. It discussed things like what font size, layout etc to try to use in your systems and what to avoid so things like needless words, long paragraphs etc. Although the book was made for mainly web design and covered a bit on mobile apps, I did feel that it was a bit outdated as it did not go in depth that much for mobile apps. Also, the examples discussed in the books are very old and websites have evolved more now.

The next resource that I found useful was a website named w3schools.com. This website allowed me to easily create a website from scratch even though I had never made one previously. The website allowed me to learn the basics of HTML code, it had categories of things you might like to add to your website so it made it very easy to understand and learn how to implement what I would want. It also enabled me to learn the basics of CSS code for the design aspect of the website, likewise it also had categories of things you would like to add to your page.

Another resource that was used a lot in my project was Nielsen's 10 Usability Heuristic Principles (Nielsen, 1994) provided from a website named https://www.nngroup.com/articles/ten-usability-heuristics/ . The resource very clearly outlined the principles with a mini description of how to avoid violating these principles, this was extremely useful as a way to try and design my interface from scratch but also evaluate it afterwards. It allowed me to follow a set of guidelines when designing my interface that I knew if I did, I would be able to create a good usable system. However, I feel one of the disadvantages of using this resource was that it only took into account general usability issues so I could have ended up finding issues according to this heuristic model however it may have actually never been an issue to the user who would end up using it.

I found I carried out a lot of research on Colour Theory when I was designing my interfaces. Colour Theory is a collection of rules and guidelines which designers use to communicate with users through appealing colour schemes in visual interfaces (*What is Color Theory?*, no date). I used an article named 'Color Theory for Designers, Part 1: The Meaning of Color' by Cameron Chapman. This was mainly used before implementing my main colour schemes into my project at the beginning, and at times later in the project. It enabled me to find out what each colour is linked to in terms of emotions and feelings. This was useful when designing for my target audiences, for example warm colours in design reflect enthusiasm and energy (Chapman, 2010) so for my GUI which was linked to being for a business I went with an orange colour as users would want to feel energy when working where as a dull colour could just make them bored. One of the issues however that I found with this resource was that it wasn't very clear and it could become a bit confusing at times, for example it would state reasons as to why you would choose to implement a colour in your design and why not. Therefore, you had to think for yourself and decide whether it be appropriate for what you are trying to implement.

Chapter 3: **Human Computer Interaction (HCI)**

What Is HCI & Why Is It Important?

This project is a study in HCI(Human-Computer Interaction). HCI is a field which studies the interaction between humans and machines. HCI mainly focuses on interfaces between humans and machines and how to design, evaluate, and implement interactive systems that satisfy the user (Mendonca, 2020). It is very important to build a good interface that a user is happy to use regularly as majority of the time they would be the client and in terms of business a poorly designed interface would not sell well. This could lead to serious amounts of money being lost out for a business. Furthermore, a poorly designed interface can have major consequences for example there have been cases which have lead to fatal accidents all down to bad design being implemented. Therefore, to build an effective system before doing anything we should set out to speak to the user and understand exactly what they want from the system. This would enable us to gather the requirements of the system that is desired, allowing us to then go about building an interface.

Principles of HCI

There are a number of principles we can consider regarding HCI such as feedback, consistency, equity, ease and simplicity (Ambielli, 2018). We are going to try and focus on these five in this project. These principles should be a way that guides our behaviour and evaluation in this project. Simplicity would link to how simple the system is to use and carry out tasks for the user. Ease would refer to how easy it is to use the design, for example is it still easy to use when your really tired, is it clear how to do things even if you aren't fully concentrating etc. Equity would link to if the interface is usable for all kinds of users. Consistency can be seen by checking if your design is implementing a similar design across all pages/windows so that a user doesn't have to re-learn how the interface works. Feedback can be linked to how the system offers the user feedback when they carry out tasks on the interface, for example if they are confirming an order, does the system inform the user it has been successful etc.

In order to try and apply these kinds of principles in my project I will be using Nielsen's Heuristic Model as I work through designing my interfaces. Nielsen's heuristic model is a well-respected model allowing you to easily evaluate your interfaces. It consists of ten principles which you should try to implement in your system and then you can evaluate at the end to check if they are included.

Issues of HCI

There are a number of issues of HCI which we can talk about which should be considered as we design a system, for example colour theory, human perception, cognitive issues, usability, design of fonts and interface problems. These issues should be considered when implementing any sort of design into our systems so we can try to eradicate them. In our project we are going to try and consider five issues which I feel are going to be most relatable to my final deliverables. These will be colour theory, cognitive issues, usability, age/target audience and interface problems.

Colour Theory is a collection of rules and guidelines which designers use to communicate with users through appealing colour schemes in visual interfaces (*What is Color Theory?*, no date). Different types of colours can be linked to different emotions and feelings for people. For example, the colour red can be linked to danger and errors (Babich, 2019) and so it could be best to avoid this as this could worry users. Cognitive issues such as memory considers not putting a strain on the

user to try and remember things to try and carry out tasks on the interface. Therefore, to try and avoid this we need to keep things relatively simple and similar to how the user would use a similar interface (Whitenton, 2013). Usability refers to how easy it is for a user to use the system and how effectively they can carry out their required tasks. Therefore, factors such as the target audience should be looked at before making the interface as we need to know who we are designing the system for, for example if we are designing a system for the older generation we would want a large font implemented so they can clearly see it. The target audience is another issue you should aim to address, as if we were to make an interface for say people who can't see the colours red or green effectively including those colours or colours that include them would be a big failure as the intended user wouldn't be able to use it. Interface problems can arise when a user perhaps makes a mistake, for example they press a button without filing the correct text fields. One way to deal with this could be to add error messaging which allows them to clearly figure out where they went wrong and carry on using the system.

Chapter 4: **Software Engineering**

Software Development Process

In terms of software engineering in my project I have made use of a Waterfall method as I worked on my interfaces. The waterfall method is a linear and sequential approach used in software development (Rouse, no date).It involves seven non-overlapping stages, when you can only move on to the next one once you have finished the previous stage. These stages are Requirements, Analysis, Design, Coding, Testing, Operation and Maintenance (Rouse, no date). When working on my project I would first set out the requirements I would like the interface to have, then I would focus on the design of pages/windows. Then I would add some code to it and finally carry out tests to make sure it was functioning as it should. This allowed me to follow a good structure as I worked on my interface and allowed me to be organized.

I opted against using other software development models such as the Agile method. The Agile method is a practice that helps continuous iteration of development and testing in the software development process. It is a process where the client is always involved with their feedback and their requirements are always changing (*Agile Vs Waterfall: Know the Difference Between Methodologies*, no date). It follows an incremental approach. As there was no real client I was producing my interfaces for I decided not to follow this method.

Testing

I made use of BDD test scenarios in my project, this stands for behaviour driven design development. Unlike TDD which focuses more on unit testing, BDD works on solving the problem of communication between the business, engineering team and the machines. Firstly, I would write a user story which would describe what a user would want to able to do with my interface. Next, I would write up a BDD scenario which would describe the behaviour the user wants when using my interface even though none of the code or design had been implemented yet. Finally, I would then work on implementing the BDD scenario into my interface and fulfilling the users requirements (Holmes, 2017). This made it easier to implement features and design as I built my interface as I was aware of what I had to implement.

TDD was also implemented in the project but only on my proof of concept program for the app. TDD focuses making good working and readable code. It involves testing specific, individual units of code (Holmes, 2017) before writing any amount of code. This test should then fail as no code has been written for it, then you would proceed to write just enough code to pass this test. Finally, you would refactor your code. I made use of NUnit testing in the program to implement TDD, I tested things such as buttons being pressed and doing the correct functionality (incrementing variables). However, I didn't make use of TDD anywhere else as I felt that it would just complicate my software development as it would be really hard trying to implement it in my interfaces.

Version Control

Version control was deployed throughout the whole of the project to keep track of all of the work being done. Version control is a system that records any changes to files over time so you can recall each version when you want (Chacon, no date). This is extremely helpful as sometimes you can make mistakes on your work and as you can always revert to an older version it saves time trying to fix your mistakes. Furthermore, the work is all stored on a repository so even if your machine stops working you can always access the work somewhere.

In my project I made use of the SVN version control system. It allowed me to store my main code in the trunk which is a folder which stores work you are generally happy with. Then when I wanted to make changes or add new features, design etc. I would create a copy of the trunk and make a

branch. Once I was happy with what I had added to the branch this work would then me merged back into the trunk which would get updated with the changes. This would be extremely useful in the real world when working in projects as this is a common system used by developers to allow them all to easily work on the same project. This is the cycle I would work in. It allowed for me to split my work up into sections and organize is better rather than doing my work all in one file. Furthermore, it allowed me to be flexible with my work as instead of worrying about whether I may break something in my program I could just focus on what I was trying to do due to the fact I could always go back to the previous version. Towards the end of my project, I also created tags which would be versions of the final main code released.

I could have made use of Github as my Version Control System which is well respected and popular however I felt due to my lack of inexperience of using it, it would just not make sense trying something different here as it could lead to the loss of my project work. Hence, I choose SVN which I have a lot of experience on as I have done a project using it before.

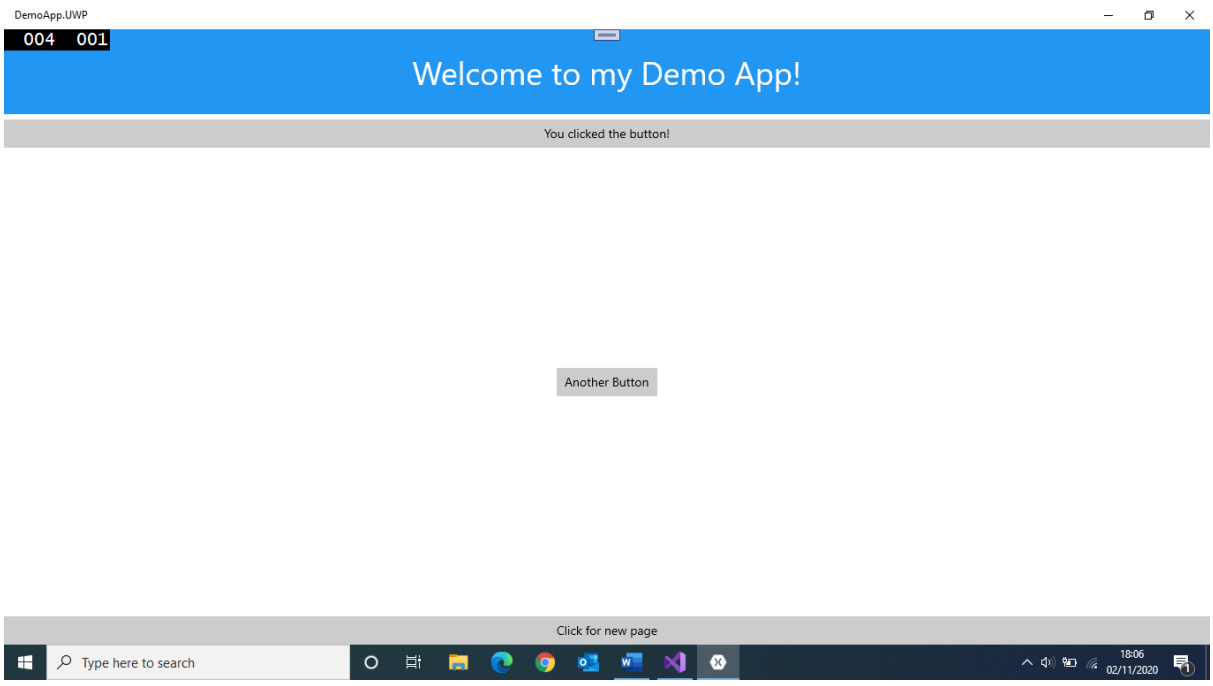
Chapter 5: **Reports On POC Development**

Android App

I've decided to make an app as one of my interfaces as a lot of people now days choose to do everything over their phones and so developing an app can help broaden my skills in the software development field, something I am interested in. Furthermore, apps offer a lot more design freedom compared to other interfaces, for example you do not have to rely on buttons to do things you could just swipe to go back.

This app made here is a demo (proof of concept) to demonstrate I have the required skills to implement my idea for the app which is a restaurant ordering system. By making a basic app like this it allowed me to familiarise myself with the basics of using a new language and software. For my demo app I made a very simple app which had three buttons as can be seen below from Figure 1, each button did something different. The first button would change the text to show you had clicked it, while the second would count the number of times it was pressed which is showed by an alert window every time you press the button (Another Button in Figure 1) and the last one would open up a different window.

Figure 1: Showing design of android proof of concept program



Alternatively, in terms of design, I could have implemented different types of clickable buttons/items as well such as clickable images, something I used in my final app. However, at the time this demo was made I was not aware that is exactly what I wanted to implement but if I did it would have saved me time in designing the final app as I would have known straight away what to do.

To create my app I've decided to go with making it on Microsoft Visual Studios as it has a developer platform named Xamarin. This meant I had to learn a few basics on the language C# which is what Visual Studios uses (*C# Tutorial - Tutorialspoint*, no date).To learn things like variable types, basic syntax and methods I made use of the 'tutorialspoint' website which showcases mini examples to help you familiarize with it. Xamarin is a .NET developer platform (*What is Xamarin? | .NET*, no date) which allows you to easily develop an app not just for android

but also iOS and Windows at the same time. This means my app will be able to work on different platforms. In addition to this as a beginner using this software is made easy as there are a number of templates you can use to make your app and a number of resources online to help understand how to develop a basic app (*Xamarin Tutorial | Hello World in 10 minutes | .NET*, no date). I made use of a tutorial provided by 'Microsoft' to understand how to get the system setup to develop an app and then how to implement functionality for buttons.

Visual Studios also has a great number of add-ons you can use for example VisualSVN which allowed me to easily connect to my SVN repository and make commits. In order to get this running and understand how to use it I conducted some research and found a website which offered the add on for SVN (*VisualSVN | Getting Started*, no date).

As I developed my demo app I made use of TDD when I added some functionality. I made use of NUnit tests which are like the standard JUnit tests to write my tests. As I had not been familiar with how to get NUnit tests, I made use of an online video (Boris Zivkovic, 2018) to aid in setting up the system and how to use it. An example of these tests was checking if the correct text would appear when a button was pressed or the correct maths was being done each time a button was pressed. These tests would originally fail and then I would write just enough code to pass them. I did encounter some issues when trying to make my NUnit tests work as there was not a code snippet that initialized the forms on each platform. To get past this I carried out some research and made use of a Xamarin package add on provided online (Peppers, 2020)which enabled me to get my NUnit tests working.

One way to assess whether the app created is following HCI principles is to link back to Nielsen's heuristic model which are design principles a system should have. Therefore in terms of design we can say that this is a rather poorly designed system as it has no real colour scheme and varying button sizes so it is lacking in consistency and standards (Nielsen, 1994). Furthermore HCI principles are concerned with making systems that are more useable so one of the goals is to try and implement an effective and efficient system (Ann, no date). While it is clear as to what some of the buttons do there remains uncertainty over what "Another Button" will do and so this would be time wasting for users if they click on it and its not what they wanted.

GUI

For one of my interfaces, I have decided to make a GUI as a lot of people use computer applications to do everyday tasks such as database operations. I choose to do a GUI over a command-line interface as they have an advantage which is that they require nothing to be memorised and they don't need to know any programming languages. Furthermore, I personally felt people would prefer GUI's as they have more graphics and are more pleasing to the eye.

For my GUI demo (proof of concept) I implemented a simple design with a Label saying "Welcome to my GUI" and a light green background. On top of this I implemented an laptop image from the internet and two buttons (Login and Help). This can be seen from Figure 2 below. The GUI has limited functionality as when you press the buttons you are just directed to new windows. When you press login, you are directed to a new pop-up page shown on Figure 3 and when you press help you are directed to a new page where you have the option to go back to the main GUI page (Figure 4). This was a very basic GUI so that I could familiarise myself with how to use the tools required to implement a GUI.

Figure 2: Showing the first page of the GUI proof of concept program

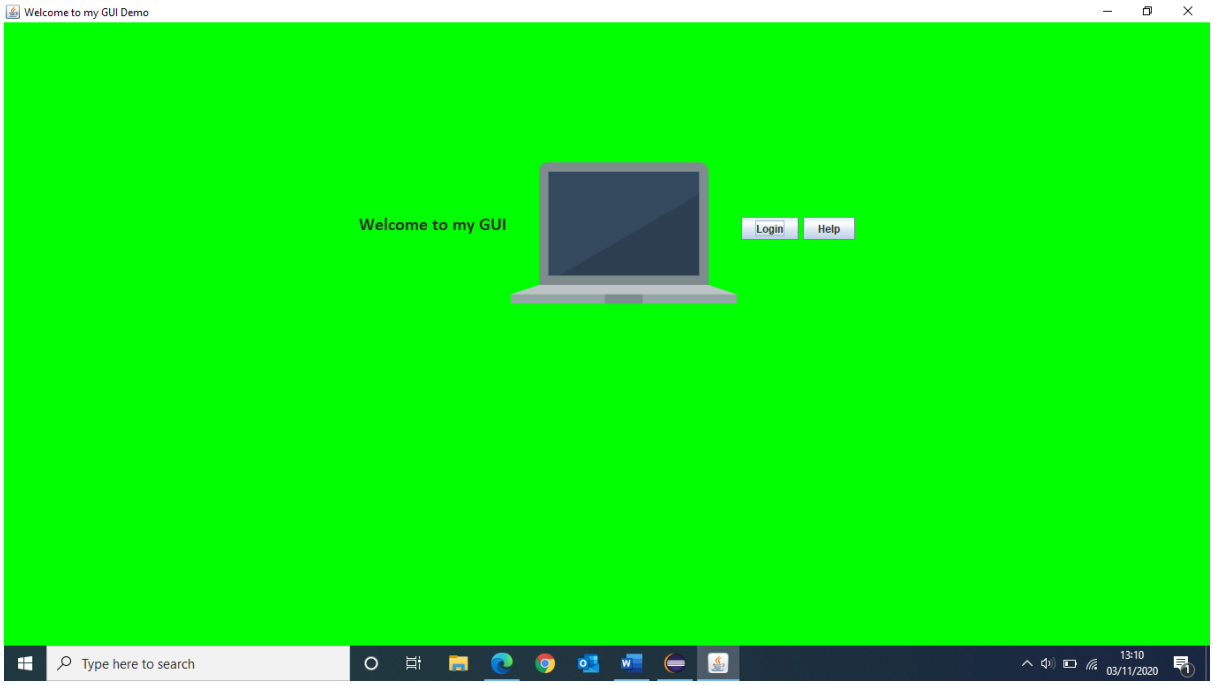


Figure 3. Demonstrates pop-up window displayed when pressing 'Login' button

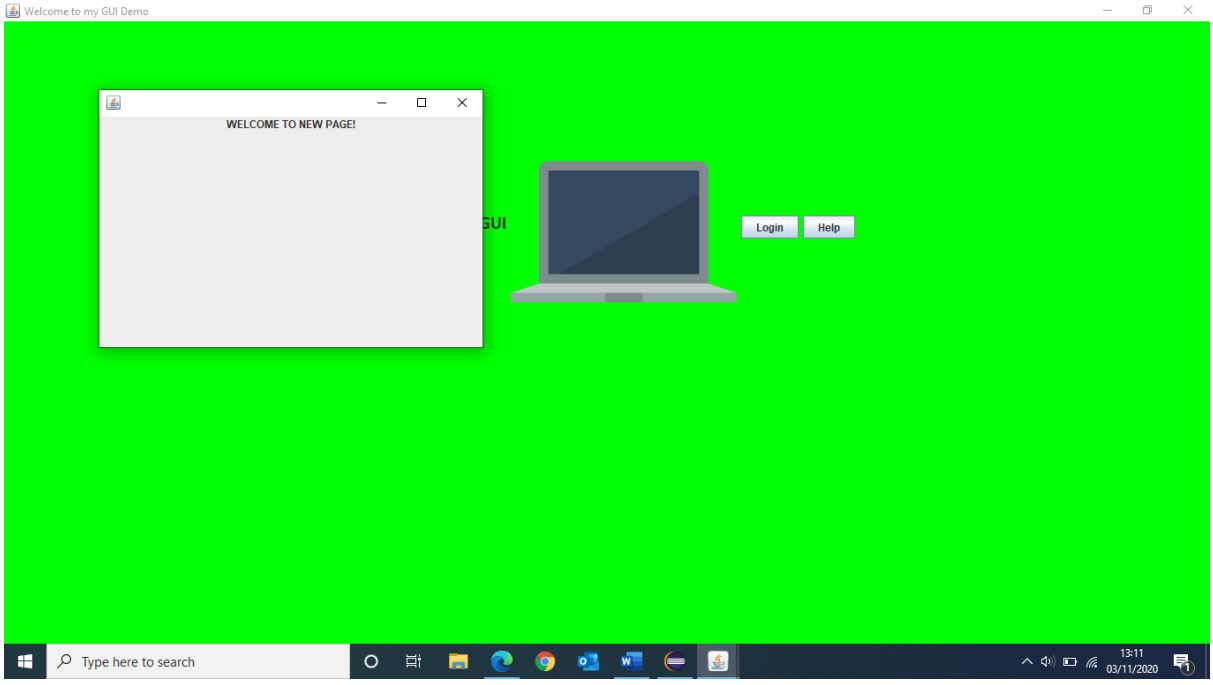
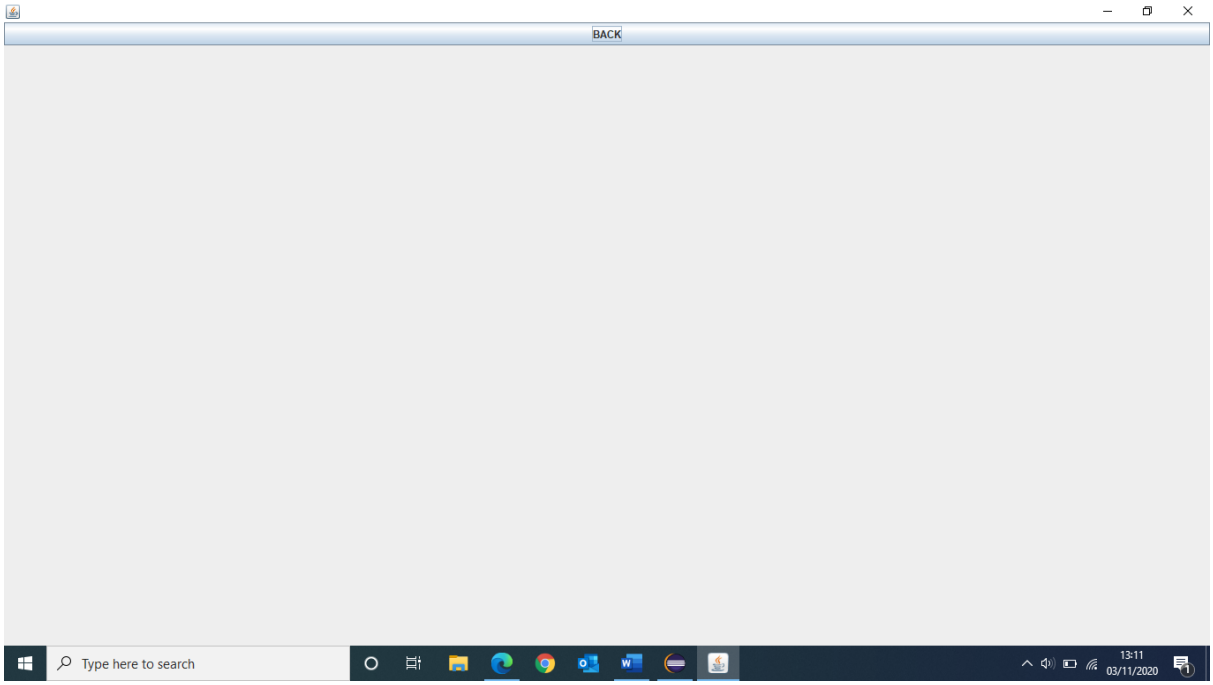


Figure 4. Shows window displayed when you press 'Help' button



In terms of design I went with the colour scheme I did because I felt a light green colour would allow for a good constrast with the black font and allow the contents of the page to be more visible.

In terms of HCI principles, the GUI made here seems fairly easy to use as it has buttons indicating where they will take you and what they will do. However, the design implemented is not consistent (Nielsen, 1994) as we have the main page with a light green background and buttons with the same size however on the other pages we have different sized fonts, no background and a large sized button. This shows it is a poorly designed interface which users wouldn't necessarily be happy to use.

To create my GUI, I made use of Eclipse which had a number of addons and toolkits such as java swing which provides an API for providing a GUI using java. Furthermore, I have used Eclipse for a long time and so felt comfortable trying to implement something different on it. To implement the GUI, I required a bit of learning of how to apply Java Swing into my project. I could have also used JavaFX over Java Swing however I choose to use Java Swing over JavaFX as swing was the standard toolkit for making a GUI for a long time so there was a lot of resources on the internet on how to implement it and it also has a bigger library of GUI elements over JavaFX (Larkin, 2020). As I began the GUI Demo development I first wrote out all the code which included the design for it with the help of online resources (Lee, 2020) however later on I found a add on called WindowBuilder (Gurubatham, 2015). WindowBuilder allows you to make a template GUI window without writing any code yourself and also allows you to easily design the GUI as it shows a design page which shows exactly what's where on the window. On top of this you can also add stuff like buttons, text etc on this window instead of writing out the code which is what I did for the windows shown in Figure 3 and 4.

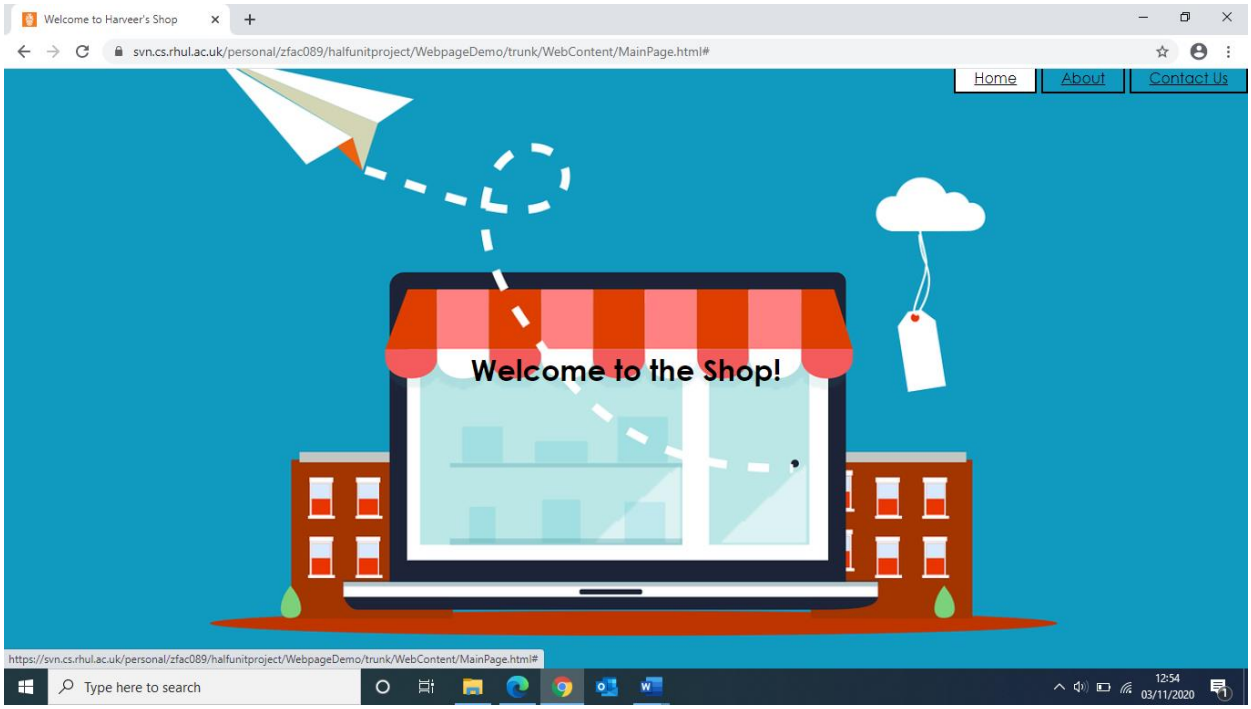
Website

For one of my interfaces, I have opted to make a website as I feel a lot of people have access to the internet and so this is a great way to advertise products. This is why my website will be to do with shopping which is an effective way businesses promote their products and make money. Also, a website is on a server therefore there is unlikely to be downtime compared to other interfaces so it

is available 24/7. I believe by getting experience in making websites now it could prove to be helpful in the future as this will help learn new skills as businesses always need help maintaining, designing websites etc.

For my demo website I made a very simple webpage which included a background image from the internet and a label in the middle of the page saying, “Welcome to the Shop!”. On top of this I included three buttons in the corner which would change colour as you hover over them. This website included no functionality and was just done to familiarise myself with the tools required to design a webpage. The design can be seen below from Figure 5.

Figure 5. Demonstrates page designed for website proof of concept program



Alternitively, in terms of design I could have gone for the navigation bar being vertical on the left however I felt with it being at the top it allowed you to have more space horizontally and allowed you to place items where you wanted them more easily.

In terms of HCI principles this seems like a very easy to use system as there are buttons in the corner which will direct you to where you want to go. In addition to this it is clear what the website is, it is an online shop which can be seen by the laptop image and the label provided. However, the label which has been included may not be as clear as it should be, this is down to the placement of it as the laptop image in the background does not really allow it to stand out and the label isn’t that big. The system does however have a consistent design (Nielsen, 1994) as all three buttons have the same effect when you hover over them which is that the background turns white.

To make my website I have chosen to use Eclipse for web development because it is something I am familiar with using so it wouldn’t take as long to get it up and running. Furthermore, Eclipse also contains the necessary tools and functionality to make a website (Zia, 2016). The languages however that is required to make a website are HTML and CSS, (*CSS Tutorial*, no date)this meant research and a bit of basic learning had to be carried out in these fields before I was able to make my demo website (*HTML Tutorial*, no date).

I had several issues trying to setup the program, this revolved around getting the correct plugins and extensions to create a website. This mainly consisted of plugins to help create a server (Singh, 2017) I could run the website on. There were also a couple of problems getting the SVN repository

working (*Configuring SVN and Eclipse*, no date) however that just required a refresher of how the plugin for it works.

Chapter 6: **Design Decisions On Final Programs**

App

Figure 6. First window of android app

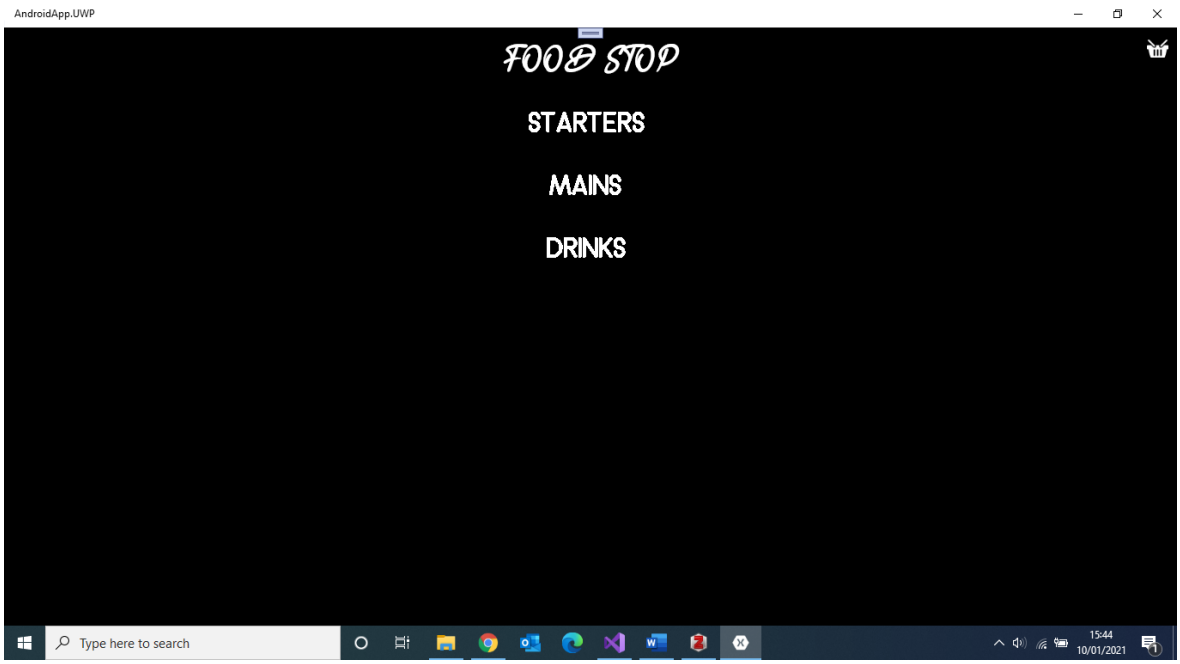


Figure 7. Example of layout shared on Mains, Starters, Drinks windows.

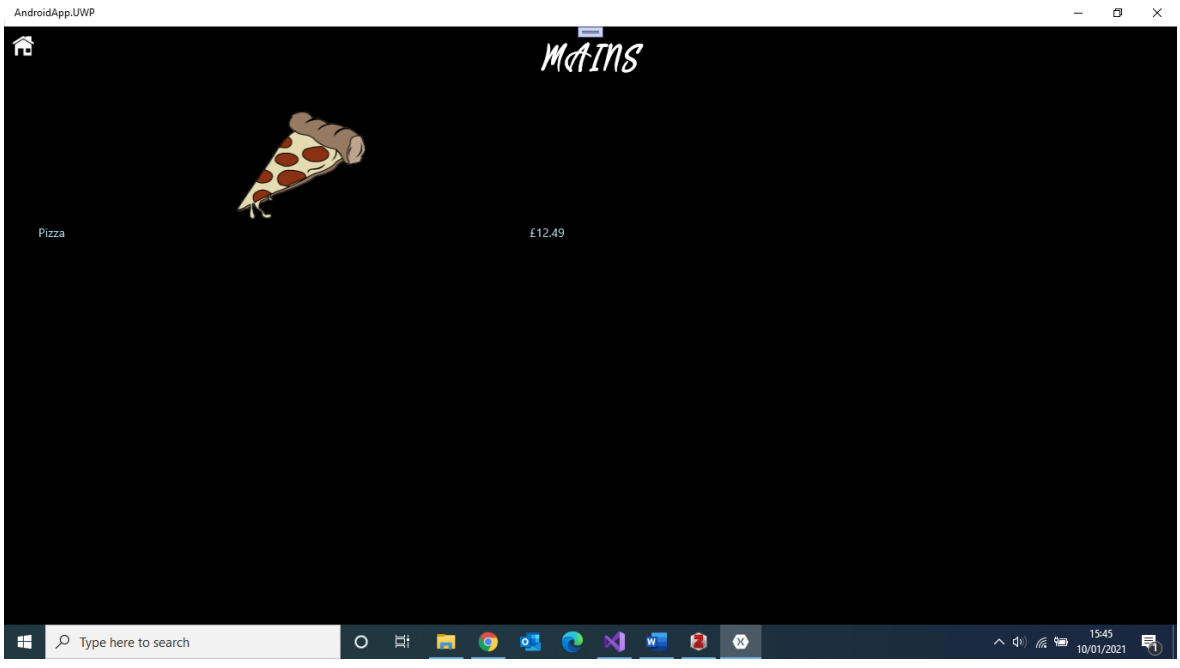
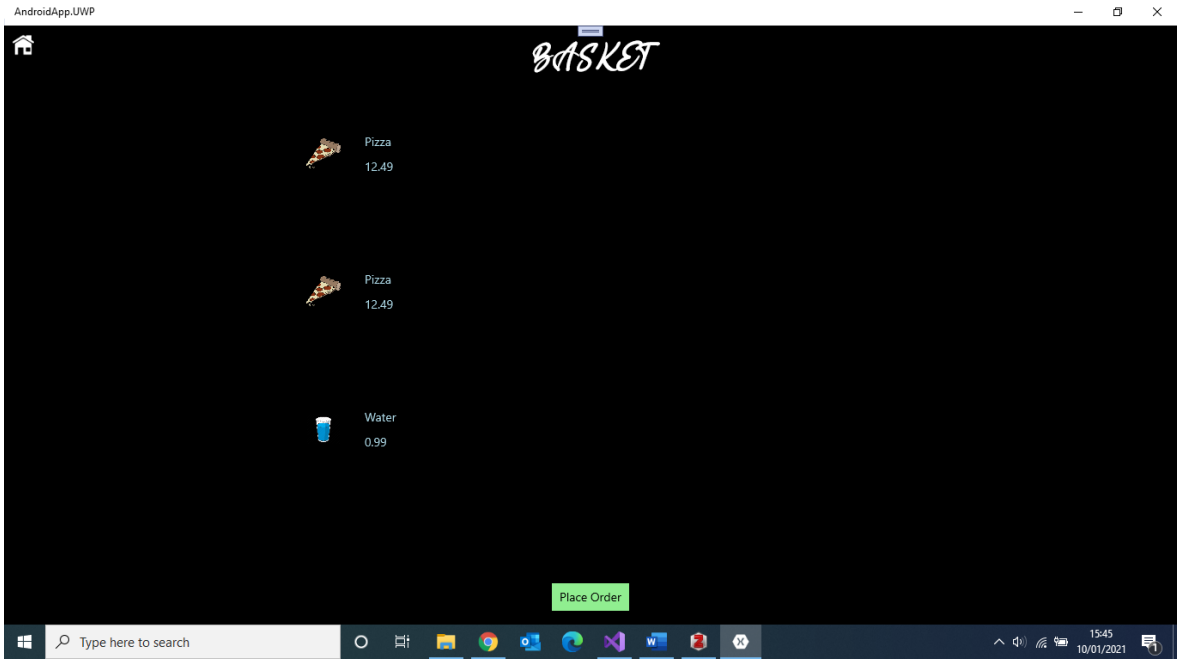


Figure 8. App basket view shown



The target audience for my app is going to be the younger generation, as I feel they would be more comfortable using an app to order their food whereas the older generation would most likely prefer to order the normal way. Therefore, I have gone with the text and icons not being that large on the pages as the younger generation are less likely to have visibility issues as can be shown on Figure 7, 8 and 9.

For my App I have decided to go with a majority black and white colour scheme with slight additions of light blue and green. The reason why I choose to go with the black and white scheme is because the colour black and the colour white are good contrasting colours and so they are easily visible (Babich, 2017) . Therefore, I implemented white on top as it would stand out on the black background. Furthermore, the colour black is often noted as a sophisticated colour and white is seen as clean, this may make users perceive the app well as it goes well to the eye (Ross, 2010).

I followed this design for all the pages as I wanted the design to be consistent as failing to could increase the users cognitive load, this is one of Nielsen's Heuristic principles. One of my original aims before beginning this project was to try follow his heuristic principles as they demonstrate good design principles. By following these principles, they are generally a sign of good user design which leads to more useable systems, a goal of HCI.

The titles on each page are separated and at the top of their pages as seen on Figure 7, the text has a bold and larger font compared to any other text on the pages, this is so that they stand out so that it is easy to grasp what the page is (Krug, 2014). Furthermore, the titles have a different font compared to the other text, this is again so that they stand out more than the rest of the information on the page and to appeal to the younger generation as it is funkier than a usual font.

On the first page of the app (Figure 6) I have decided to go with clickable text buttons which direct to the starters, mains, and drinks. On the Windows App version it is also clear these are clickable as when you click them a slight effect occurs where the text goes a tiny amount smaller and depresses in (Krug, 2014).This again reduces cognitive load on users allowing them to use the interface quickly without much thought. This is aligned in a vertical manner and spaced evenly from each other. I went with this as it separates the products and makes it even easier for a user to figure out what they want to find as if it was cramped together it would be hard to press the items. This is

linked to making it easier to use the app for the user, a key element of Human Computer Interaction.

In addition to this, I have tried to add a rather simple design as users don't want to read much as they most likely would scan the pages and these clickable buttons stand out quickly as they clearly state what they will redirect you to (Krug, 2014).Furthermore I went with a clickable basket button in the corner of the page as this is a standardised way people implement the basket checkout on other websites, apps etc. This can be linked with one of Nielsen's heuristic principles (Nielsen, 1994), which is to have a match between the system and the real world so straight away without even thinking many users would look to the corners to try and access the basket.

On the Mains, Drinks and Starters pages (Mains page can be seen on Figure 7, all other pages similar to this) we have gone with a home button in the left-hand corner, this is the exact same reason as with the clickable basket button in the corner discussed above (match between system and real world). On these pages the products have a rollover effect when you hover over them on the Windows App demonstrating they are clickable and on the Android App version they turn orange when clicked, this is important in showing they are clickable which again reduces cognitive load on users. These pages also have the text around the images showing the price and name of the product in the colour light blue as it is perceived as providing security to people, and this is important as it is demonstrating price details which could decide whether a user wants the item or not. Therefore, by hopefully providing security I'm hoping the user will feel safe that this is a reasonable price (Bayston, 2020). This will hopefully increase sales, something a business would want. These pages also have images as this is what most online restaurant apps have so there is again a link to the real world.

The Basket page (Figure 8) included a green "Place Order" button at the end so that the user can review the items above it before clicking this. Also the colour chosen here was green as it is linked to excitement and new beginnings, which is feelings linked to when people finally confirm their order and wait for their food (Cherry, 2020). The new beginning link can be linked to the fact it allows you to make another order after pressing it as well. Again, on this page it is easily identified this button is clickable as it shares a rollover effect and is in the shape of a typical button. On this page we also have a view of the ordered items which can be removed, and so this allows for user control and freedom (Nielsen, 1994) to undo mistakes which have been made.

GUI

Figure 9. Demonstrates first window of GUI

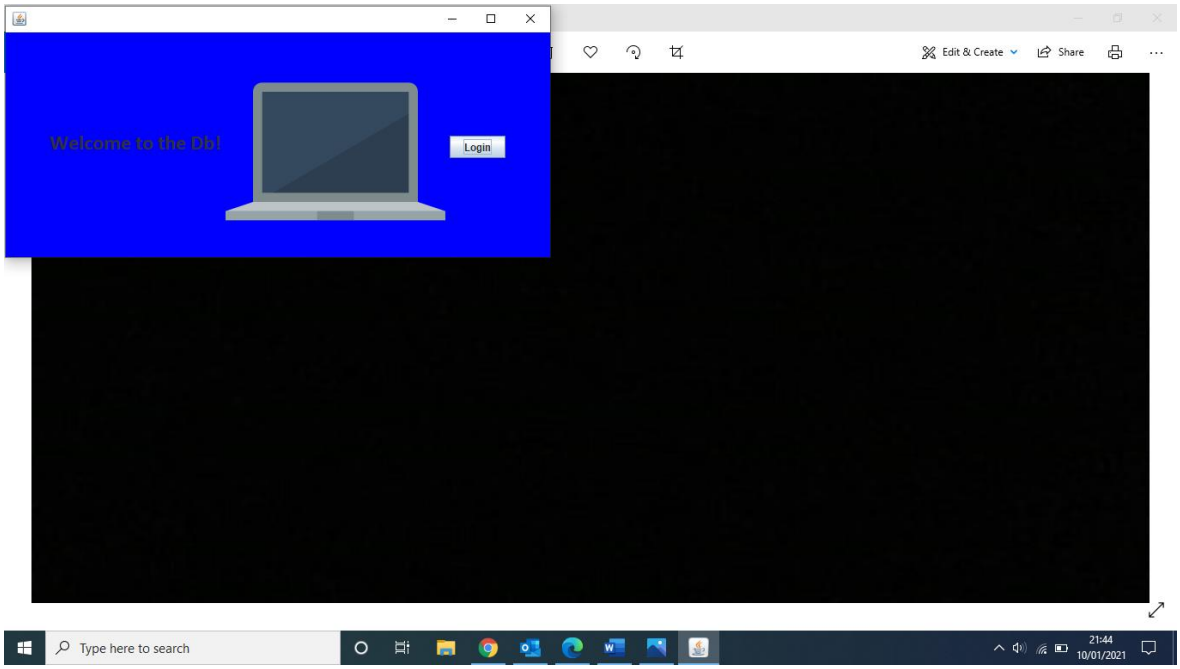


Figure 10. Login window displayed when you press 'Login' button from Figure 9

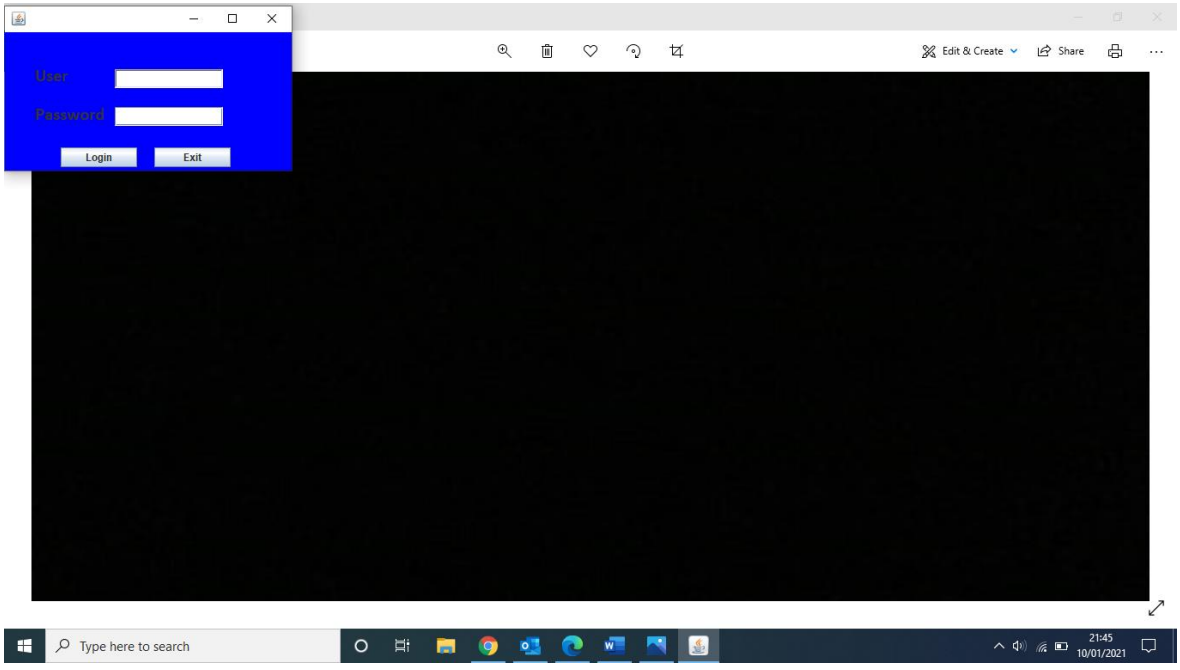
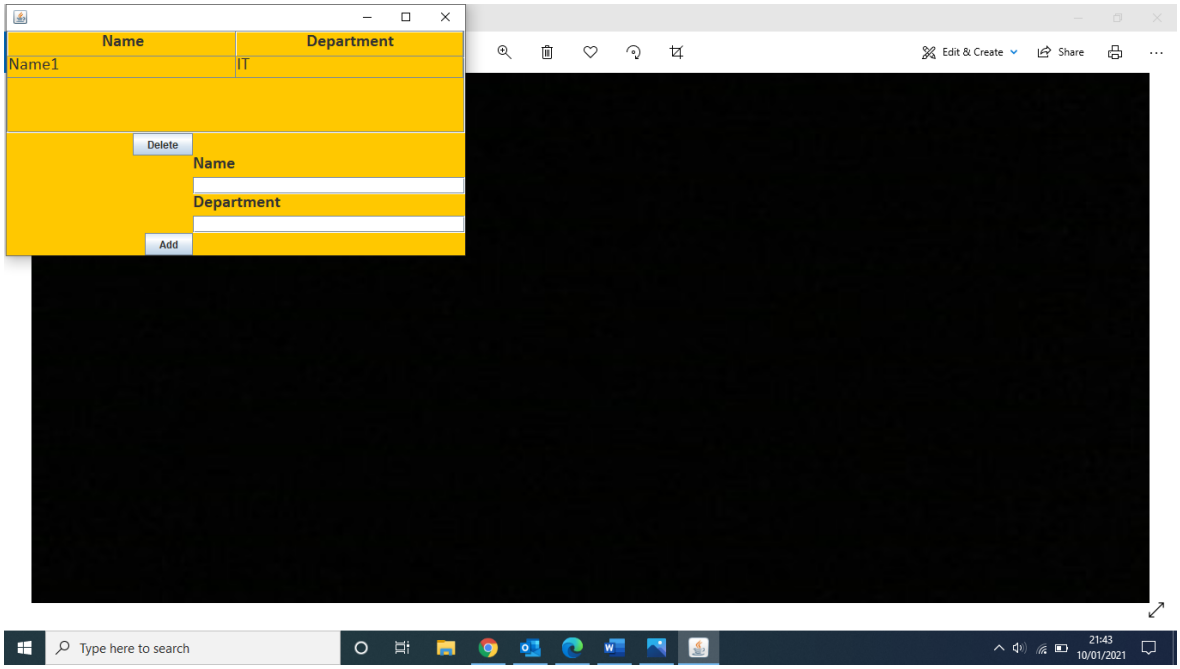


Figure 11. Main database window shown once login is successful



The target audience for my database GUI interface is going to be the working population as companies are most likely to use an interface like this. As the working population can cover a wide range of ages, I have tried to adapt my interface so that it can accommodate the majority of them. For example, I have made the headers in the interface larger than normal and with bold fonts so that it is readable by people perhaps in their 50s or 60s whose eye sight may not be the best. I also made my text size on the database screen (Figure 11) larger than average to get past this issue.

The colour scheme in my interface here involves a blue screen before we login (Figure 9) and a orange screen (Figure 11) once we have logged in. The reason why I have gone with blue is that it is considered a colour which promotes sense of trust which is important in this scenario as you are going to be asked for your login details (Figure 10). Then when on the page where all the work will be done we go for orange because it boosts creative performance and generates enthusiasm, something which a company would like from their employees (‘The Best Colors For Productivity And Creativity In Your Workplace’, no date). I have gone with black text on all the windows as it provides a good contrast for visibility and so reading will be made easy (Babich, 2017) which is vital as without good visibility a system isn’t really usable.

Buttons were added on the windows as they were to show clearly they were clickable hence why we used a different colour. I could have been more creative here and changed the colour from the standard given one however every other colour I tried didn’t look professional, something we want from a GUI going to be used by a business. Also, I felt it wasn’t necessary and it would only cause confusion. These buttons also consisted of a border to clearly show it is a button. This would reduce cognitive demand on the memory of the user into trying to figure out whether it was a button. This is one of Nielsen’s heuristic principles (Nielsen, 1994). One of my original aims before beginning this project was to try follow his heuristic principles as they demonstrate good design principles. By following these principles, they are generally a sign of good user design which leads to more useable systems, a goal of HCI.

I have tried to keep a match between the system and the real world as well (Nielsen, 1994), this can be demonstrated by my design on the login page. This is another of Nielsen’s heuristic principles I tried to implement when designing this window. I went with a familiar kind of login page others would have possibly experienced on other GUI’s or websites. This means by implementing the

design I did for this window it would reduce cognitive demand as without even thinking a user would feel comfortable using the GUI as they would have most likely experienced something similar and so little thought would go into what to do.

I have tried to add a simple design as users do not want to read much when working, they would just want to get to the required page (Figure 11) as quickly as possible. Furthermore, it is one of the heuristic principles to have an aesthetic and minimalist design (Nielsen, 1994). I did this by not putting any sentences, paragraphs etc on any of the pages, just the necessary information as a user is likely to scan the pages (Krug, 2014). Furthermore, the clickable buttons used would also stand out quickly as they clearly state what they will do for you (Krug, 2014) allowing for a user to quickly navigate through the GUI.

The layout for my first page (Figure 9) can be seen to be horizontal going from left to right, I went with this layout over my original idea which was to have the Login button under the laptop image however this would mean not being able to apply a layout manager. This meant if my user was to make the screen larger it would not adapt to the user's screen and would stay in the same position as it were when the screen were smaller. Unfortunately, I could not figure out a way to apply this to the Login window (Figure 10) without making it look unprofessional.

Another reason why I kept a lot of my text implemented in the GUI of bold font is so that they stand out as this is the information I want the user to easily grasp on the page (Krug, 2014), for instance on figure 11 I've kept the labels near the text fields for Name and Department bold so that they will most likely see this, and if they see it they are less likely to make mistakes like input the wrong information.

Website

Figure 12. Main page window of website, similar layout is implemented on other pages

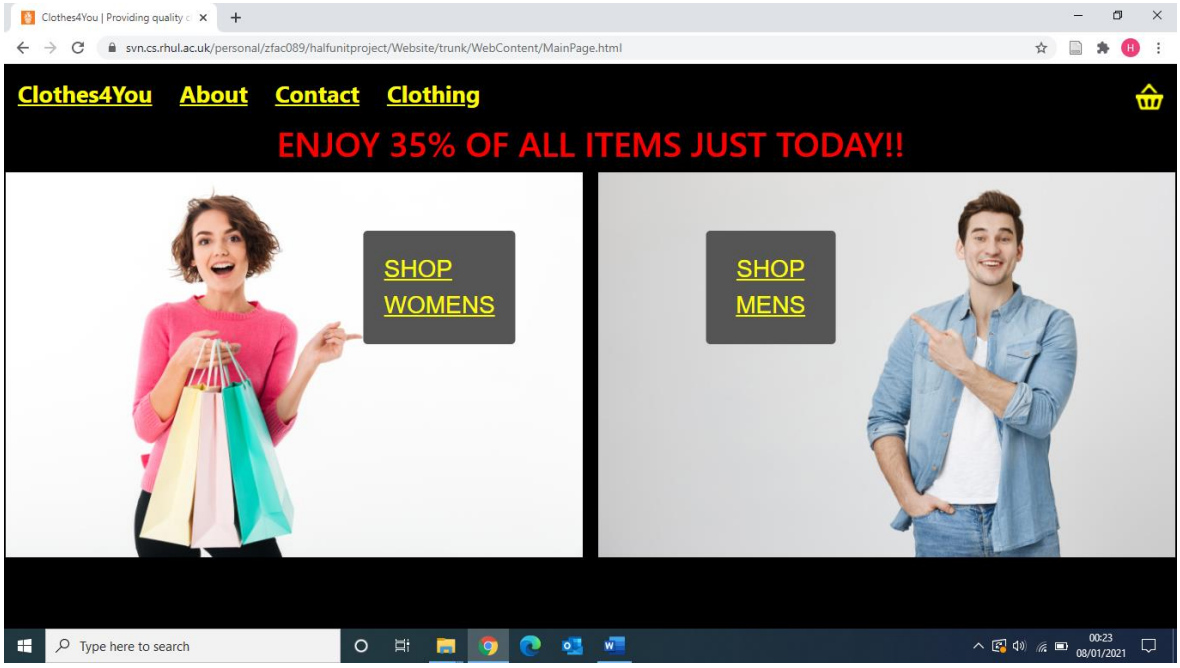
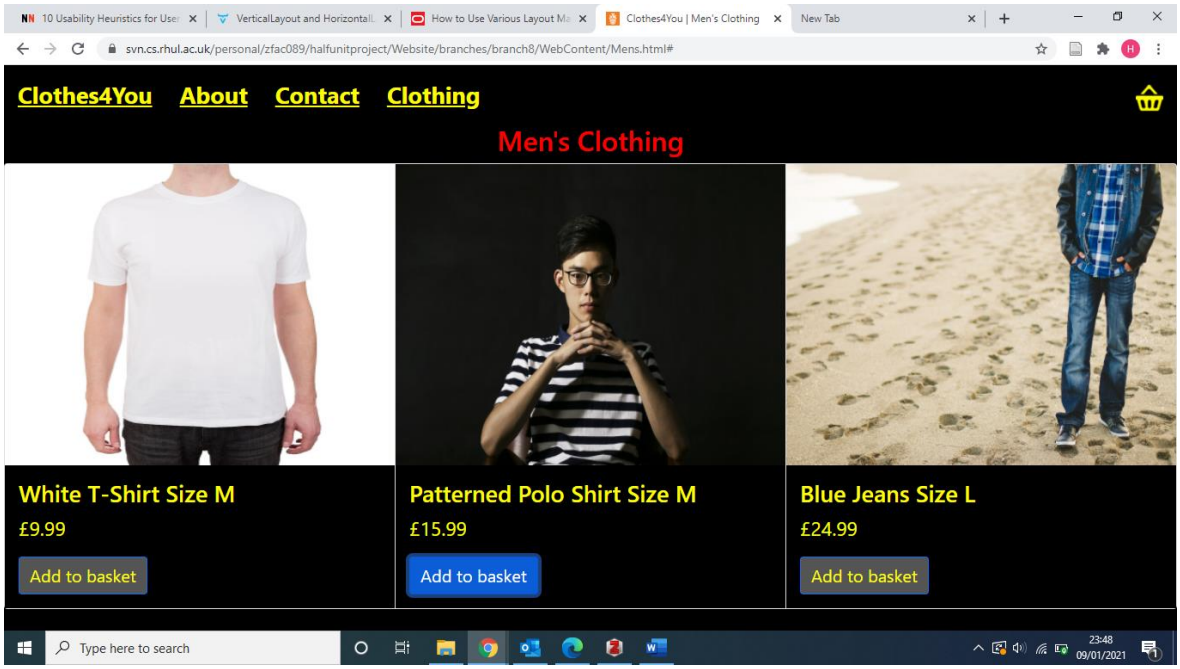


Figure 13. Demonstrates Men's Clothing page (same layout as Women's Clothing)



My website was made with the intention of being for those that can't get out during this pandemic to shops to buy clothes, this would be those that are considered a high risk to go out. This means I was mainly targeting the elderly population as they would be the majority in that bracket. This is why I have tried to make everything on the page as large as possible, for example the basket icon. This is down to the fact that as you get older you tend to find it harder to see smaller things. This is why I also included two photos of people pointing to a button to press for the desired gender shopping as shown in Figure 12.

The colour scheme implemented on the website consists of majority black and yellow colours. This is because these are both contrasting colours and so it makes reading things a lot easier, this is a good thing due to the fact my target audience is the elderly. The colour yellow was chosen after finding out that elderly people tend to prefer yellow colours (*Renk Etkisi | The Effect of Color | Color Preferences in the Elderly*, no date). I didn't want to choose too many colours as well as it has been found that using combinations of colours make it harder to differentiate from each other if they are used in close proximity (DeRemer, 2015). Furthermore, the colour yellow is linked to comfort and energy (Ross, 2010) which is vital as you want your users to be comfortable using the website and have energy as this is an online clothing site so hopefully it can help draw more sales.

The colour red is used in the header titles on each page, I choose this because I wanted the title of the page to gain attention so it is clear what the page is and also to get across important information (e.g. the sale title in Figure 12). In addition to this, red is one of the most common colours used to increase sales (Morris, 2013), something a business would desire.

A navigation bar is present on all the pages on the top as I feel it gives you quick access to pages which you may find helpful such as contact. The navigation bar consists of underlined names to clearly show that these are links that can be pressed. I decided to add a basket icon in the corner of the page as this is something you would see on the majority of online shopping websites you access, this reduces cognitive demand on your memory. This is good as we want the elderly to be able to easily use the interface without having to think much at all. The fact we have things like a basket icon and a navigation bar on the top shows a match between the system and the real world (Nielsen, 1994). This is one of Nielsen's heuristic principles, something which I want all my interfaces to try and meet. By following these principles, they are generally a sign of good user design which leads to more useable systems, a goal of HCI.

I could have gone with a vertical navigation bar instead of a horizontal navigation bar. My first ideas consisted of a vertical navigation bar as I designed my website however I found that having a vertical navigation bar made the canvas smaller for my images and as I wanted them as large as possible so that the elderly could easily see it, I decide to opt against it.

I have tried to keep the design consistent, so all the pages will be following the same black and yellow colour scheme. Again, this is one of Nielsen's heuristic principles. This would mean users would get a feel for what the system does on each page, as different layouts would cause them to have to learn again how the page works, this would hopefully reduce cognitive demand on the user. I have also added a rollover effect to the buttons where they turn from yellow to white, this helps in making the user aware they are hovering over the hovered button. The title of the page has also been made larger so you can easily grasp what page you are on as that allows it to stand out (Krug, 2014).

The buttons used in my website consisted of a grey background with yellow text as I didn't want to keep it all black and yellow as I felt the user would end up feeling bored. Furthermore, I still wanted to use the yellow font text as the most important thing to me is that the user can see what is written so they know what the buttons do, and the grey colour was high in contrast compared to yellow so it allowed the yellow to stand out. The grey colour also has shades of black which was my background colour, as I didn't want to make it so different to this it proved to be the perfect colour. Although the grey colour is linked to being dull and emotionless (*The Color Gray*, no date), I opted to use it over other darker colours such as brown due to it looking more professional.

The buttons for the Men's Clothing (Figure 13) and Women's Clothing pages (same layout as Figure 13) have a slightly different effect when you press the Add to basket button, once clicked it becomes blue with the text turning white. I felt that this could serve as a remainder for the user that they have already clicked the button and so they won't make the mistake to press it again. This tackles one of the Nielsen's heuristic principles, it prevents error prevention as this colour change of the button could prevent a user pressing the button again. A white border layout is also used on the Men's and Women's clothing pages for the items to demonstrate separation from each other and avoid confusion.

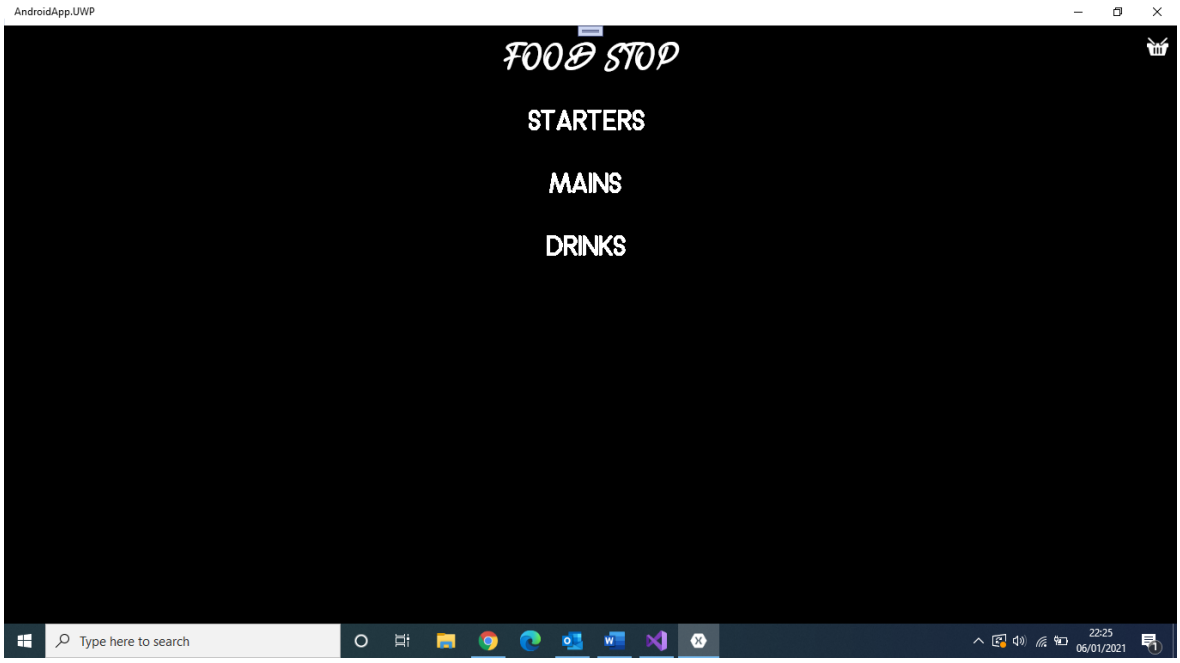
Images were used throughout the website as this is something you are most likely to see on other websites and so again it meets one of the heuristic principles of their being a match between the system and the real world (Nielsen, 1994).

Chapter 7: **How To Use Programs**

App

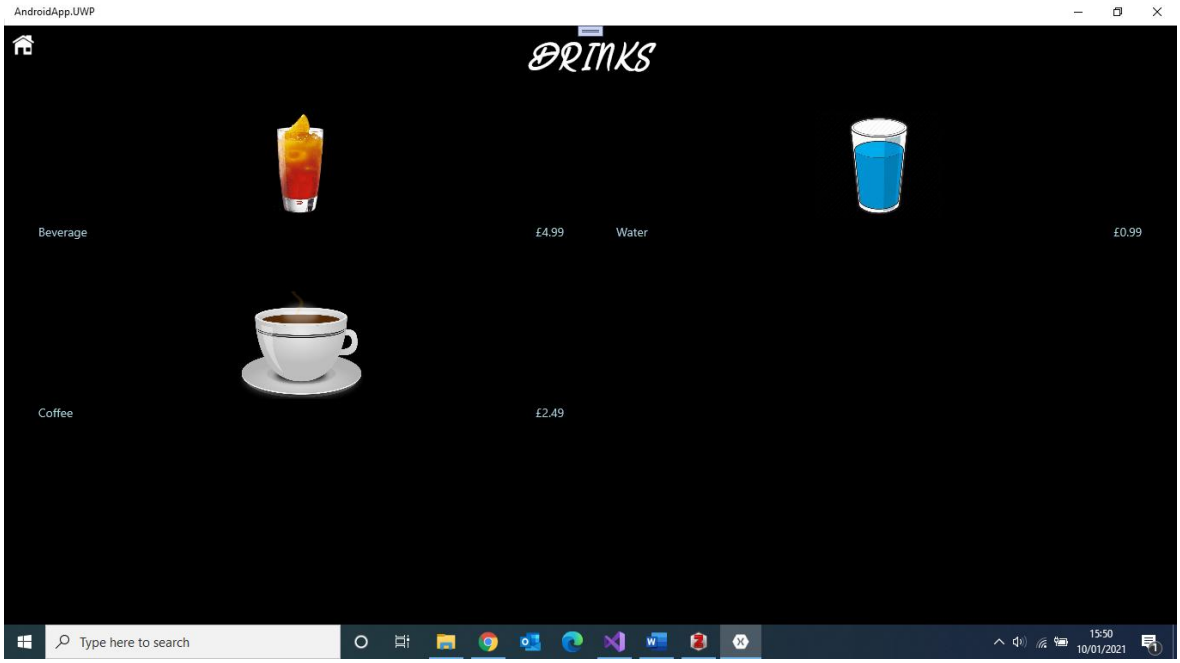
The app can be used on either Windows (UWP) or Android. To get it to run you must install visual studios and load the solutions in. Once you open the app you will see the main page consisting of the types of food you can access and a basket icon in the far right which will direct you to the basket page when clicked. This can all be shown below on Figure 14.

Figure 14. Demonstrates initial screen shown when app opened



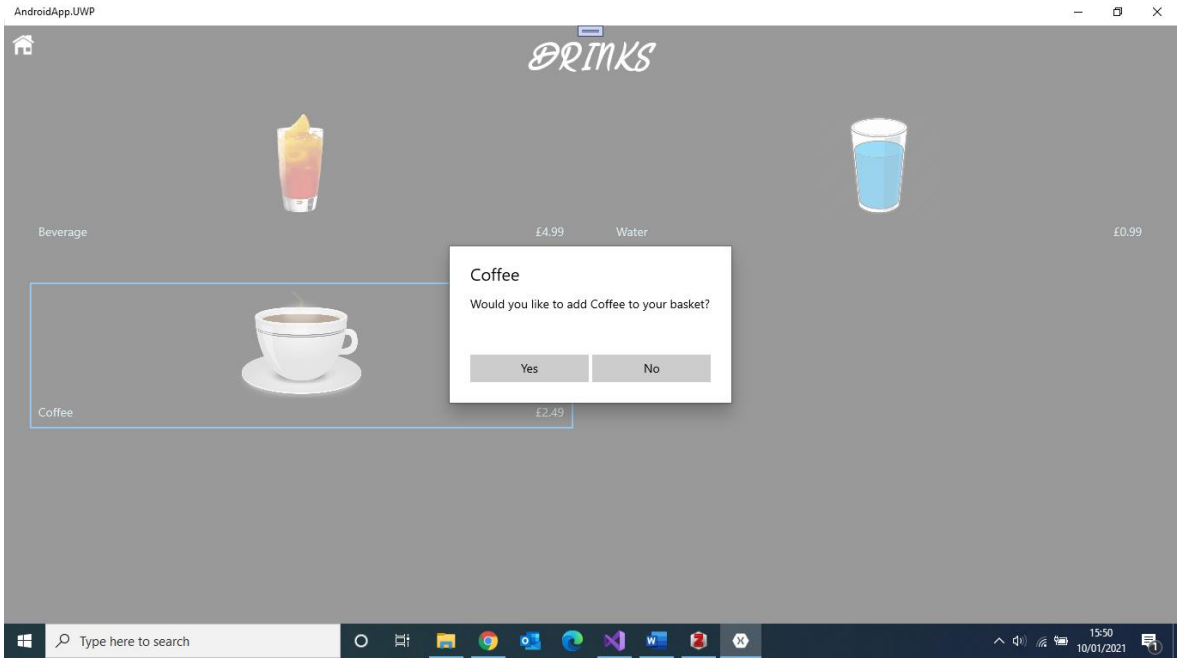
To order food you can click on one of the types of foods, for example if you would like Drinks you click on that which will take you to a new page as shown in Figure 15 consisting of different types of drinks you can order.

Figure 15. Shows drinks menu(same layout is shared by mains and starters as well)



Then to order the items you click on the image of the food which will bring up a pop-up message asking you if you want to add it to your basket as demonstrated by Figure 16.

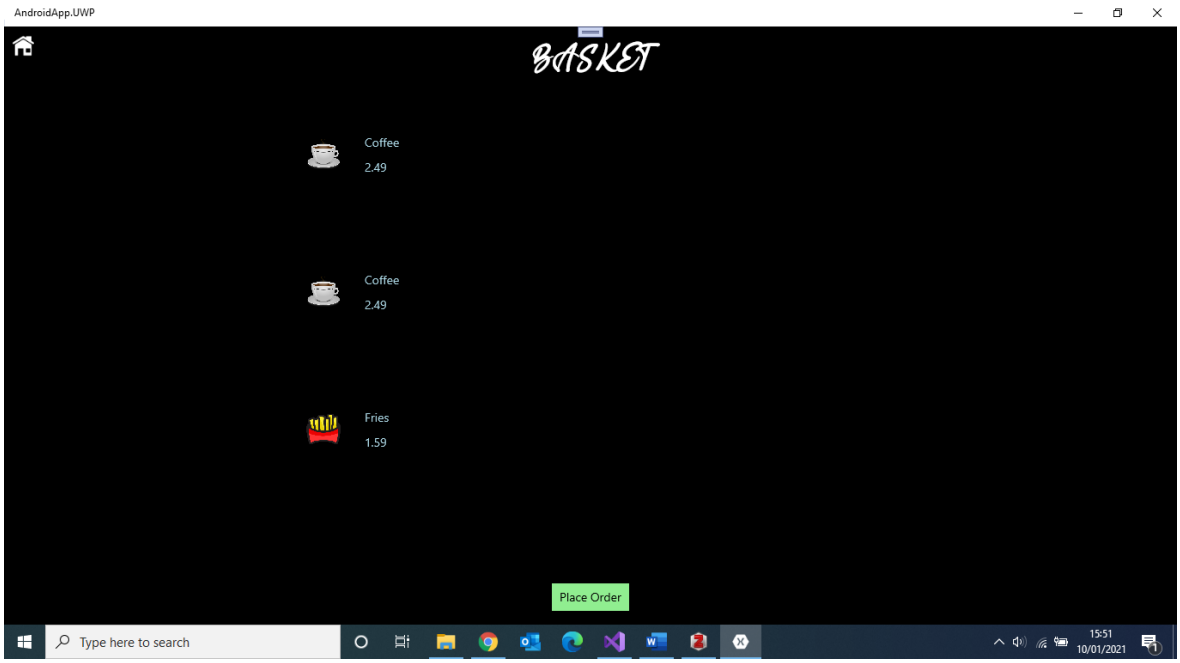
Figure 16. Demonstrates pop-up message shown when item clicked



If you click yes, it is added to your basket. Once you have pressed Yes or No, the pop-up message disappears and you are left with a view of the items again (Figure 15) where you can order more of the same or different products from the page. If you would like to view your basket or order items from a different page you can click the home icon in the top left corner which will take you back to the first page we saw (Figure 14). From here you can direct yourself to the other food pages and order food the same way as discussed previously or view your basket by clicking the basket icon in the top right corner. All the other pages on this app will also have the same home icon in the top left corner enabling you to go back to the original home screen page (Figure 14).

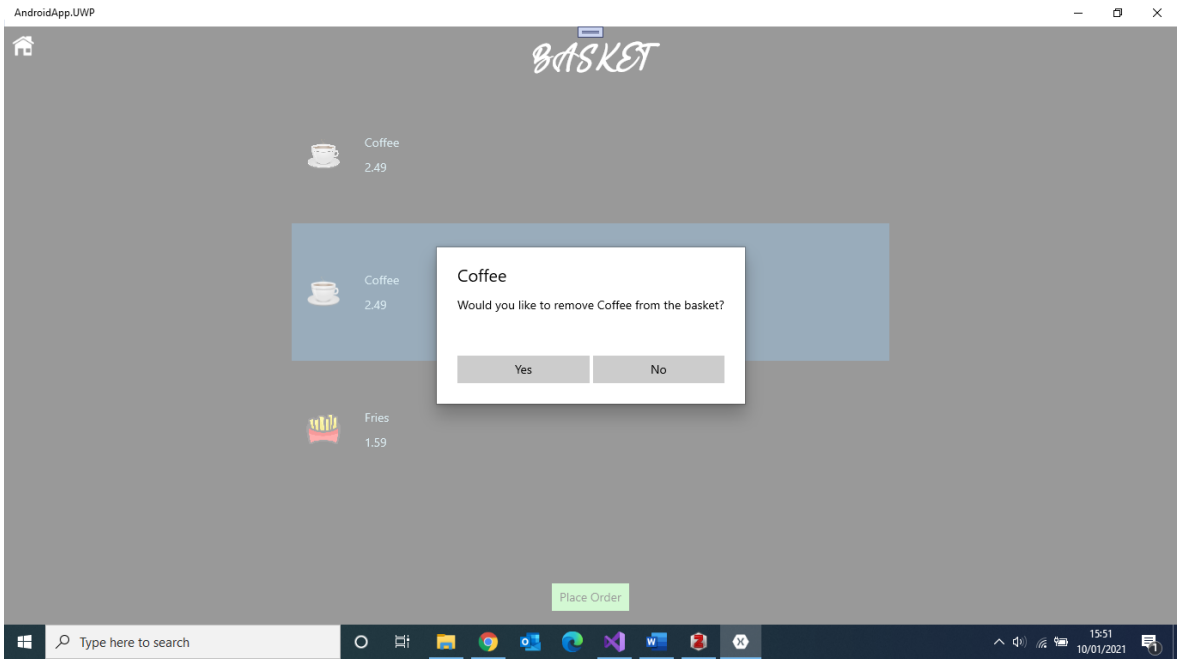
When you do decide to press the basket icon from the original home screen page, you are directed to a page consisting of all the items you have ordered. This is shown in Figure 17.

Figure 17. Basket view shown when you click basket icon from figure 14



If you click the basket icon without having ordered any items, no items will be displayed. From here you can also change your ordered items by deleting them if you decide you no longer want them. You do this by clicking on the item you no longer want, then you are presented with a pop up message asking if you would want to delete your item (Figure 18).

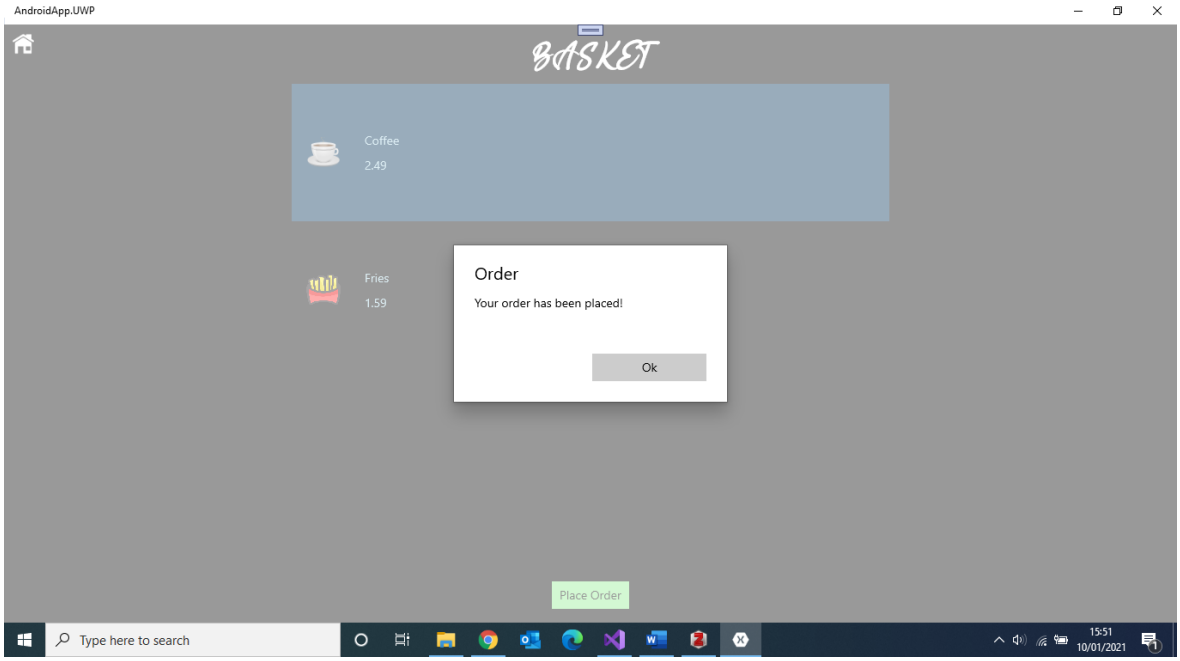
Figure 18. Pop-up message shown if you click on item in basket view



If Yes is clicked, the item is removed from the list. Once you are happy with your order you click the green button at the bottom “Place Order”, this will alert you with a pop up message if your order has been successful as shown in Figure 19. Then you press “Ok” to this message which will

take you back to the original home screen (Figure 14) and from here you can make another order if you like as the items from your basket will reset and now be empty again. If there are no items in your basket and you click the button you will be told that there are no items in your basket.

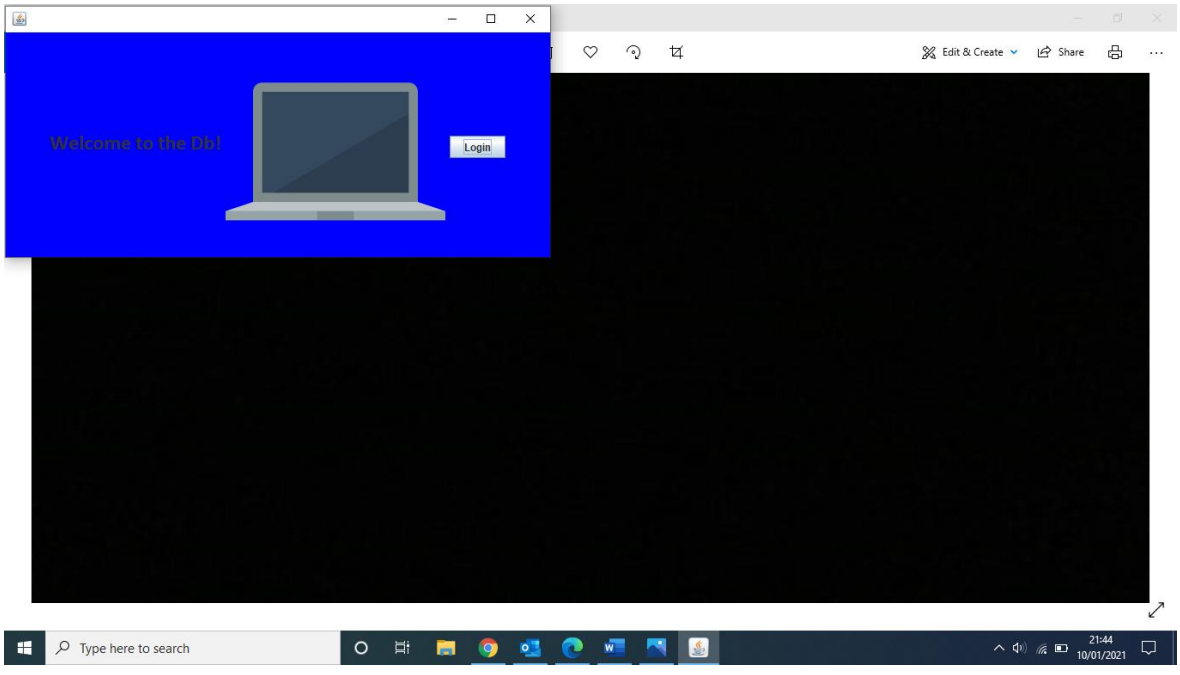
Figure 19. Pop-up message displayed when you Place Order with items in your basket



GUI

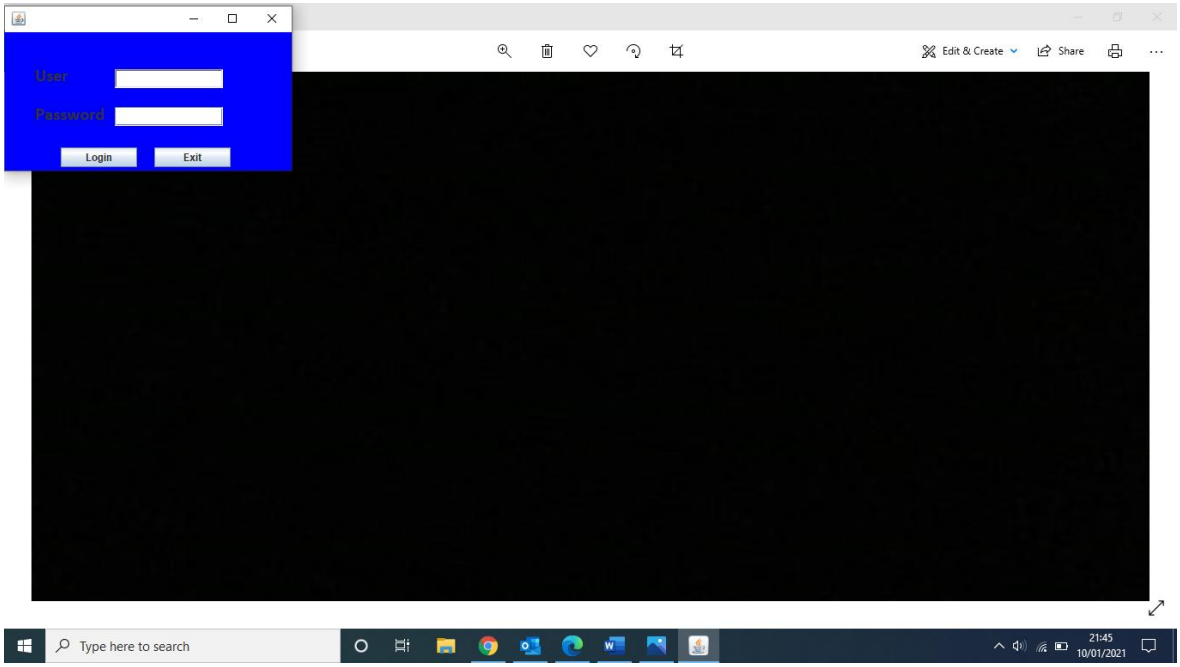
The GUI is a Java run program so it can be used on computers. You run the program by running its Jar file which loads the first window on the GUI. Once open you are presented with a window as shown in Figure 20.

Figure 20. First window displayed when GUI is run



When you press the login button you are redirected to a login page where you must input your username and password. This view is seen on Figure 21.

Figure 21. Login window displayed when 'Login' button clicked from figure 20



You can also click the back button which will take you back to the original main window (Figure 20). The username is 'user' and the password is 'password' to login. Once inputted you should be notified whether or not you have logged in successfully, if not you can try again, once you have pressed Ok to the pop up message. If everything has gone correctly you press Ok to the pop-up message saying you have logged in as shown on Figure 22, and you will be redirected to a new window consisting of a database (Figure 23).

Figure 22. Pop-up message demonstrating login has been successful

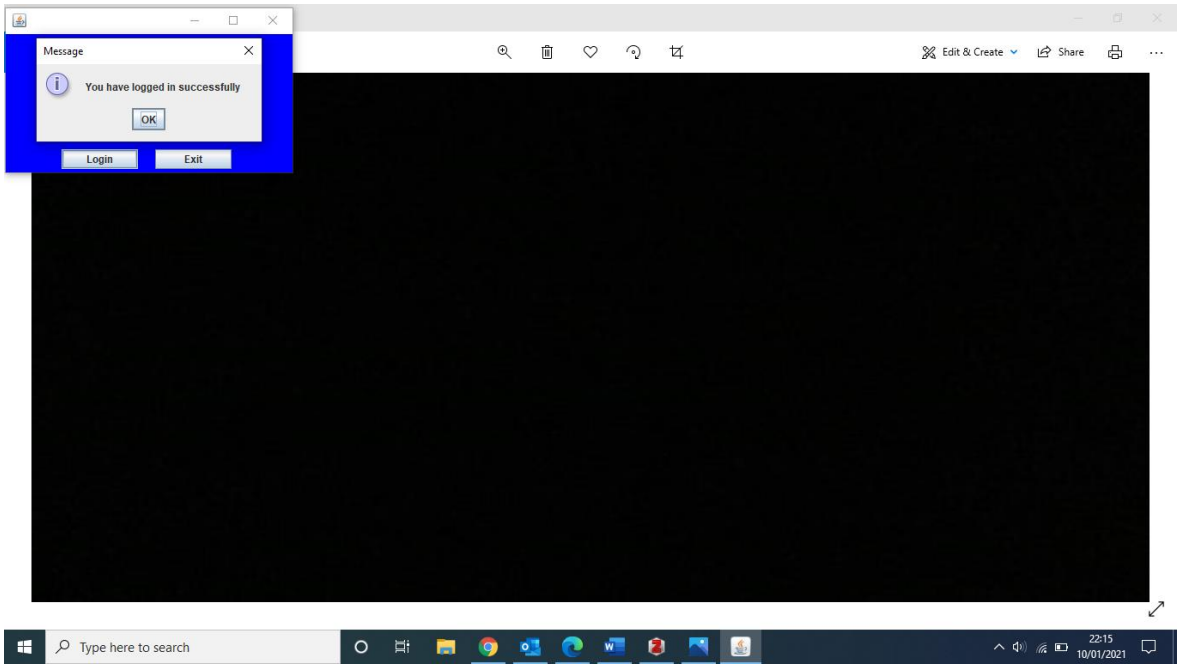
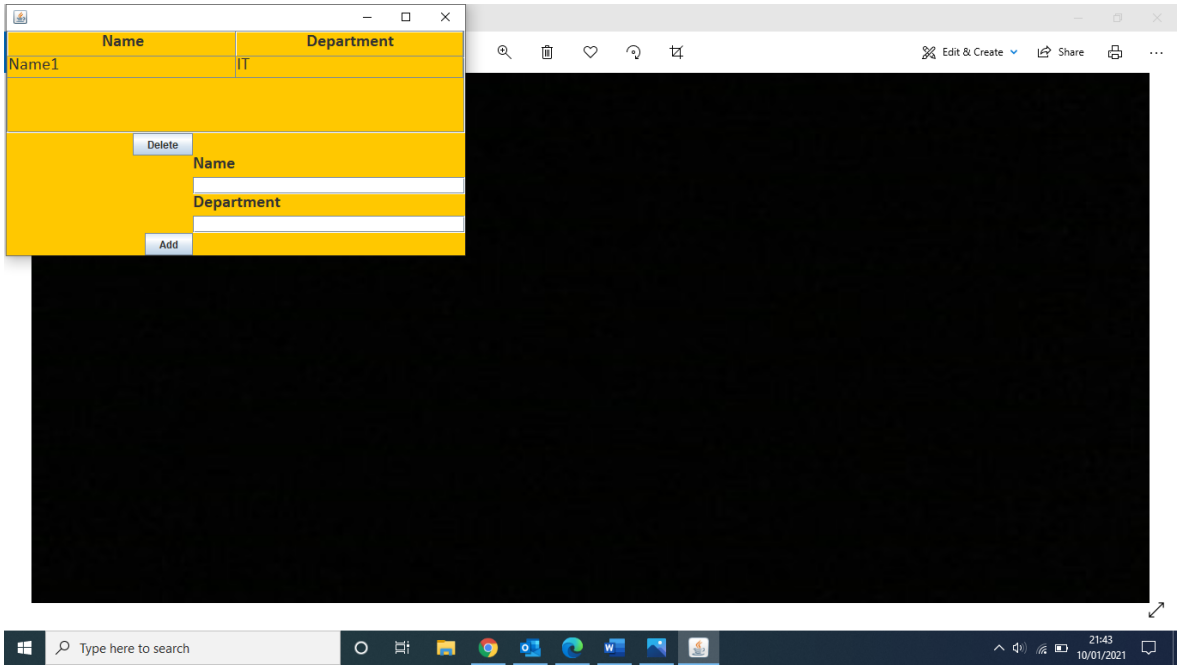


Figure 23. Database screen shown once login successful

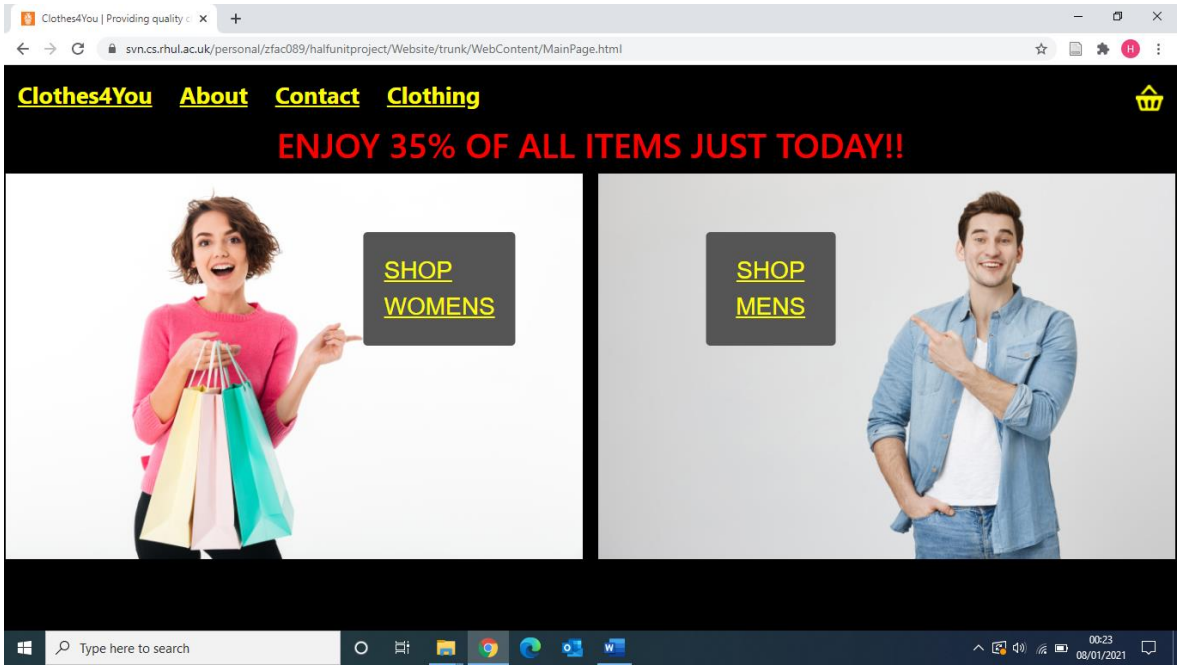


From here you can view the data already in the database and add/delete data as you like. To delete data, you click on the row on the table, this will highlight the chosen row and hit the delete button. If no chosen row has been selected and you hit the delete button you will be notified with a pop-up message to select a row. To add data, you must enter in the Name and Department text fields and hit add which will automatically populate the data onto the table. If either the Name or Department fields are empty you will be notified when you press the add button by a pop-up message.

Website

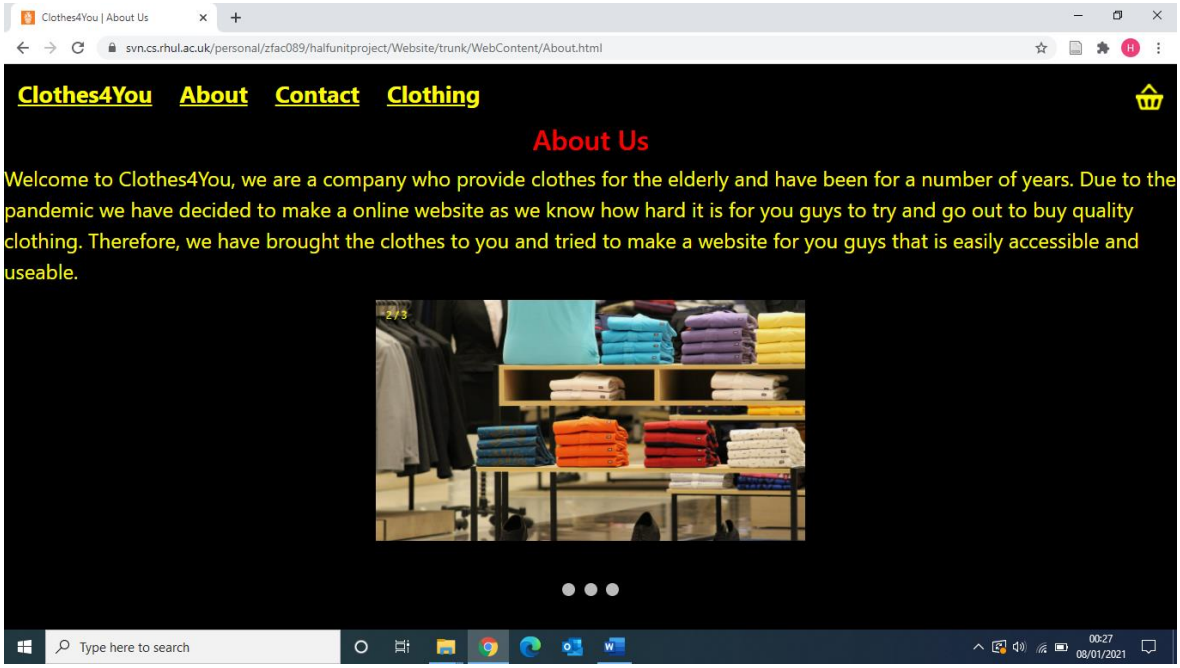
The website is a HTML script and can be used on anything that has internet access. It is run by opening the MainPage.html file. Once you access the website you will be greeted with the main page which consists of a navigation bar (available on all pages) at the top consisting of Home, About, Contact and Clothing which is a dropdown link consisting of Mens and Womens if you hover over it. All these links will direct you to their pages. On the far right we also have a basket icon which will direct you to the basket page. The layout can be seen below on Figure 24.

Figure 24. Demonstrates layout of main page on website



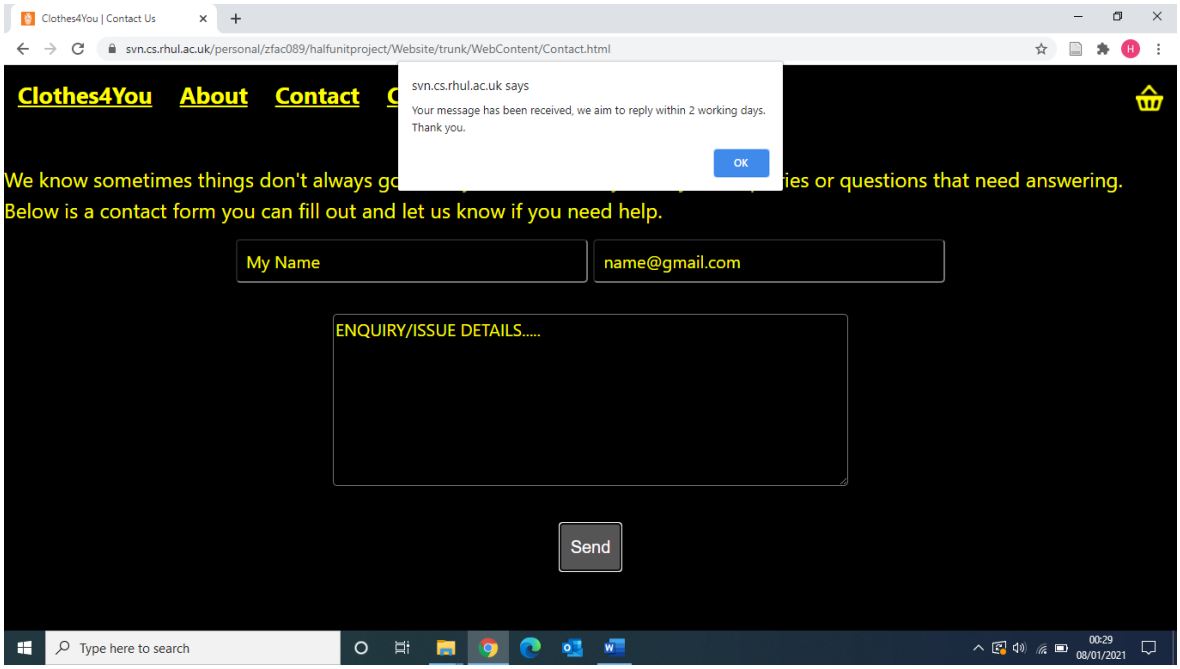
The main page also consists of 2 images and buttons on them. You can also go the Mens or Womens page of clothes via the main page by clicking the buttons shown in Figure 16 on top of the images. The About page will direct you to a page consisting of a page with information about the company and a slide show of images (Figure 25).

Figure 25. About Us page, accessed by pressing 'About' from navigation bar



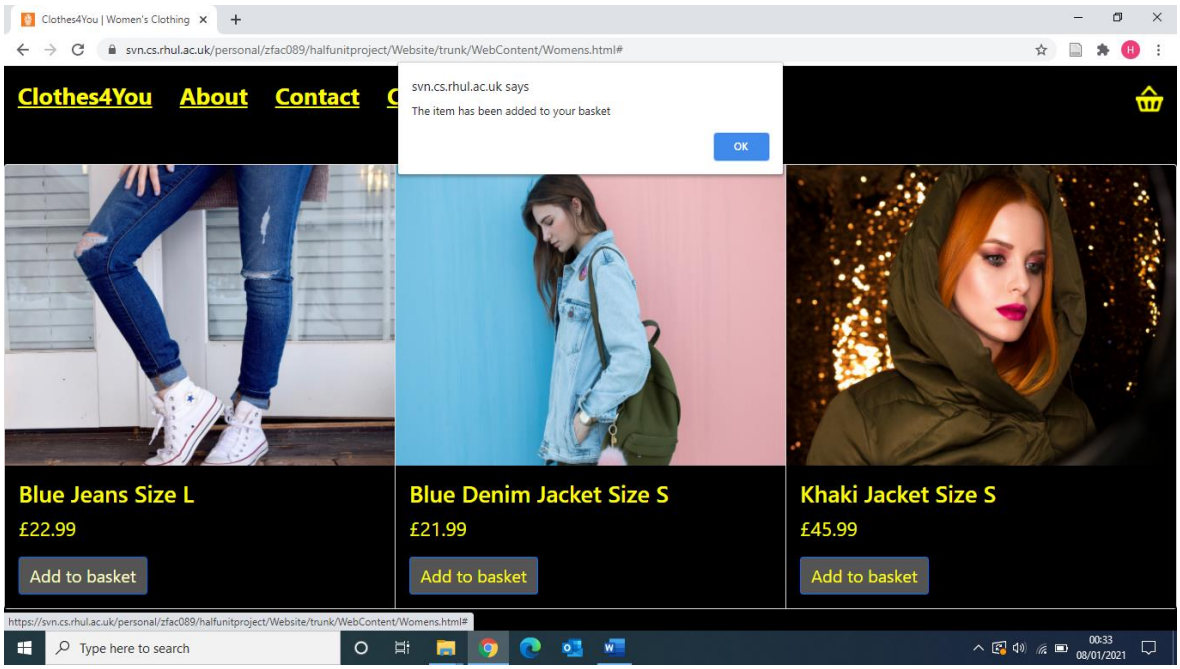
If you direct yourself to the Contact page you will be presented with a form to fill out if you would like to enquire about anything or have any issues. Once you have filled out this form you click on the Send button at the end and you should be greeted with a pop-up message indicating that the message has been sent as demonstrated in Figure 26.

Figure 26. Pop-up message shown on Contact page when you click Send



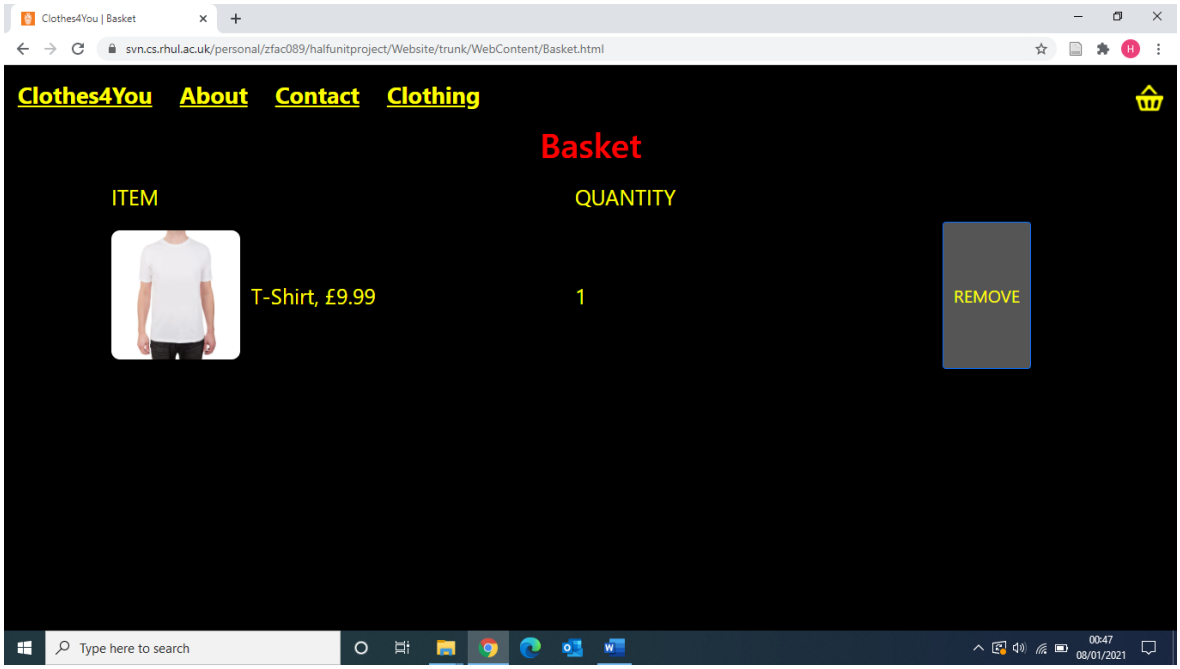
To view items and buy products you can access either the Mens or Womens section, in Figure 27 we access the womens section. Both pages run the exact same. To buy a item you would simply press the Add to basket button which will present you with a pop up message indicating the item has been added to the basket.

Figure 27. Women's Clothing page demonstrated with pop-up message when you click 'Add to basket'



The basket page accessed from pressing the icon in the top right corner shows a T-Shirt always in your basket due to the fact functionality was never implemented for this as I ran out of time. Therefore, this is just a demo to show how it would be. The page allows you to view products in the basket and has a remove button if you no longer require it as shown below on Figure 28.

Figure 28. Basket view, accessed by pressing the basket icon in the top right corner



Chapter 8: **Evaluation Of Three Interfaces**

We are now going to evaluate the 3 interfaces made and compare them in terms of human usability. Usability will be used evaluate them because it will tell us how well the user can use the system, which relates back to HCI as we want to build a system that satisfies them. Usability can be compared via principles of HCI such as feedback, consistency, equity, ease, and simplicity.

Both the app and website share a consistent design as they have the same colour themes implemented on each page, and everything behaves similarly in terms of links and icons so its easier for the user to use the system as they don't have to learn how new things work. On the other hand, my GUI has a different colour scheme on the main database page compared to its other pages and a slightly different layout so this could confuse users and make them work harder to use the system as they will have to figure out how things work again. Therefore, by not implementing a consistent design for the GUI I have not followed HCI principles here. If I was to carry on with this project it is definitely something I would consider changing, so for example using one colour scheme throughout and a standard layout.

Feedback can be seen on the app and GUI as when you make mistakes you are told why the action can't occur and are offered advice on how to avoid this, for instance on the GUI if you login with incorrect credentials you will be notified that either your username or password is wrong. This is useful because it notifies the user where they have gone wrong, if this wasn't the case the user would become confused and frustrated as to why they can't carry out their desired action. This would make the system quite unusable which would be a bad thing as if the user can't use the system effectively, we would have a poorly designed interface. The website offers feedback in some parts by telling the user their action has gone through when they click on buttons however it does not offer advice if you do something wrong. For instance, on the contact page if you forget to fill out a field, let's say your email and you send your enquiry by hitting the send button you are still alerted with the same pop-up message indicating that the enquiry has been sent. However, it wouldn't be possible to send an enquiry without an email so if I had more time to spend on the project, I would add some error handling here.

Simplicity is demonstrated through all the three interfaces I have created. This is because all the systems are easy to understand on how to use without even needing a user manual regardless of how well the user understands technology. Furthermore, the interfaces only include information that is necessary for the purpose of usability and so users won't feel like what they are viewing is pointless. The website is kept even more simple compared to the GUI and app due to its target audience being the elderly, so we have tried to include clear clickable things on the pages as they are less likely to be up to date with technology, for example all links are underlined demonstrating they are links that will take you to a different page and no fancy effects are used.

Ease is referred to how easy the design is to use even if a person is not fully concentrating or is tired(Ambielli, 2018). In terms of ease, I would say the website would be the most easy to use as it has larger fonts than both the app and the GUI. The GUI wouldn't be the worst as the text is also larger than average on most pages however my app consists of the item name and price being relatively small and this could turn out to be a problem as even though the target audience is young people if they are not concentrating or are tired it could lead them to disregard that. Therefore, if I had more time, I would most likely get a group of young people to test using it while tired and see if they picked up on the information provided from this font and then made changes if required.

Equity is shared by some parts of the interfaces as to have equity an interface would have to be usable by different types of people (Ambielli, 2018). For example, with my website I have included bits of red colour which would not be usable for people who suffer from being protanopia colour blind (Lockett, 2020). If more time were available, I would have tried catering for this and perhaps other types of colour blind by carrying out research as this would make my app more useable.

However, the app does include a black and white colour theme throughout which are contrasting colours and so it does allow for easy visibility for the majority of people. The website should be usable by quite a few people as it has large fonts which would prevent people with visibility issues (regarding size) to be able to use it. In addition to this, the yellow colour scheme is mainly deployed with a black background to add contrast allowing for easier visibility on the website. The GUI also shares some elements of equity, for example the colour theme of orange and black allows for contrast and so again it improves visibility.

Chapter 9: **Self Evaluation & Professional Issues**

Self Evaluation

The project went well throughout, I feel as I was able to stick to my plan originally made at the beginning of the term for the majority of the weeks which allowed me to work at an steady rate and not just do a lot of work in one week and not the other. Therefore, I feel my time management in this project was good. In terms of developing my proof of concept programs, I feel I should have maybe spent a bit more time on them and developed more skills working on them as when I started my final programs I still felt I had to research how to implement the things I wanted even though I knew the basics. I felt I should have probably practiced what I wanted to implement in my final programs on the proof of concepts instead of just the basics of the new languages and software's I was using.

The project allowed me to gain new skills and experience in using new software's/languages I had not used before, so I believe this can help me in the future as employers will look for people with more experience. By learning this new software, I believe I will try to use these more as I found them quite fun to use as well. I learnt that as I developed the interfaces it allowed me to figure out whether I truly enjoyed software development. I found that it was relatively fun the majority of the time and it is most likely going to be a field I am interested in for a career in the future.

Professional Issues

The topic I am going to discuss regarding professional issues is licensing. Licensing involves permission to use or own something, it represents an agreement between someone who wants to use the work and someone else who will give permission to use it (*Licences and Rights*, no date). In terms of this project, it would be to do with using things from the internet. It is a massive issue as by not having any rights to use something it could lead to bad consequences such as you breaking the law and having perhaps a court case against you. There are a number of issues regarding Licensing such as whether or not work is open source licenced. Open sourced work is work that can be freely accessed, used, changed and shared by anyone (*Frequently Answered Questions | Open Source Initiative*, no date), this means it avoids the plagiarism issue. Plagiarism is when you pass of ones work as your own without giving credit to them in any sort of way, this is an ethical issue which can lead to your work not being accepted and can be considered as breaking the law. To avoid this issue it is best to cite your sources and present your own ideas (Calonia, 2019).

As I developed my programs, I found this to be an issue because I found that as I was creating interfaces on both languages and software's which I had not ever used before it would be hard to avoid using bits of code from other people's work. In order to avoid this however I would draw inspiration from what I saw from other people's work and use basic syntax and tutorial websites to try and understand how I could implement my own ideas. Furthermore, if I felt I had to use someone else's code in my work I would reference in my code that it was not mine and where it had come from.

In addition to this, I found that as I developed my interfaces, I wanted to use images in them to make them look more appealing. This proved to be a major issue as with some images you can't use them without paying for them or just down to the fact people don't want their work being used by other people. This meant that I had to pick my images very carefully and read all the small prints on them about use. This led to more extensive searches on the web to try to find the correct kind of images I was looking for. I was mainly looking for copyright free images as these are free for anyone else to use or images which clearly stated that you could use them for your own personal use. I found that some images were also allowed if you would clearly reference them in

your work so this was also another way around the licensing issue, therefore I referenced the images in my code to show where they came from.

Chapter 10: **Conclusion**

In conclusion I believe the three interfaces I have created are perfect for the study in to HCI as all of them tried to deal with the issues of HCI such as colour theory, cognitive issues etc. On top of this, they all shared principles of HCI which would normally lead to a good system being developed for the user. Before beginning this project, I also had a aim of trying to implement Nielsen's Heuristic principles into my project, I wasn't able to achieve all of them however I implemented a few throughout the project. This would lead to a well-designed user interface which I hope I have implemented.

However not everything went according to plan throughout the project, for example I felt my time-management perhaps was not the greatest towards the middle of my project development. This was because of other coursework's I had during that period. Furthermore, I felt I took too long developing the app due to issues with implementing the basket/checkout view. This led to my original timetabled plan not being followed around after week 6/7 and less time being spent on the other interfaces, especially the website. This led to me not being able to finish the functionality I originally had planned for the website. In addition to this, I did not have time to test all my interfaces on different types of screens/devices to see if they held their layouts. Therefore, if I had longer to spend on this project, I would have carried out more tests on my interfaces and spent longer on them to add more features. I would also have made more complex interfaces, for instance with my app instead of relying on buttons to navigate pages I would implement it using swipes.

Bibliography

‘A Guide to Human-Computer Interaction (HCI) | Adobe XD Ideas’ (no date) *Ideas*. Available at: https://xd.adobe.com/ideas/principles/human-computer-interaction/ (Accessed: 18 November 2020).

Agile Vs Waterfall: Know the Difference Between Methodologies (no date). Available at: https://www.guru99.com/waterfall-vs-agile.html (Accessed: 5 January 2021).

Alex Lee (2020) *Java GUI Tutorial - Make a GUI in 13 Minutes*. Available at: https://www.youtube.com/watch?v=5o3fMLPY7qY (Accessed: 30 November 2020).

Ambielli, B. (2018) *15 Principles for Human Centered Design*, *Bambielli’s Blog*. Available at: https://bambielli.com/posts/2018-08-06-fifteen-principles-for-human-centered-design/ (Accessed: 1 December 2020).

Ann, K. (no date) *What is Human Computer Interaction (HCI)? | by Ann Kim | Medium*. Available at: https://medium.com/@annjkim/what-is-human-computer-interaction-hci-3020e5c29e5b (Accessed: 30 November 2020).

Babich, N. (2017) *The Underestimated Power Of Color In Mobile App Design*, *Smashing Magazine*. Available at: https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/ (Accessed: 4 December 2020).

Babich, N. (2019) *Using Red and Green in UI Design*, *Medium*. Available at: https://uxplanet.org/using-red-and-green-in-ui-design-66b39e13de91 (Accessed: 1 December 2020).

Bayston, R. (2020) ‘6 Colors That Are Proven to Boost Sales’, *The Daily Egg*, 21 March. Available at: https://www.crazyegg.com/blog/colors-proven-to-boost-sales/ (Accessed: 10 January 2021).

Boris Zivkovic (2018) *Unit testing in Xamarin.Forms - Programming Addict*. Available at: https://www.youtube.com/watch?v=_GsWot-Ndk4 (Accessed: 30 November 2020).

C# Tutorial - Tutorialspoint (no date). Available at: https://www.tutorialspoint.com/csharp/index.htm (Accessed: 30 November 2020).

Calonia, J. (2019) *How To Avoid Plagiarism, How to Avoid Plagiarism: 5 Easy Methods | Grammarly*. Available at: https://www.grammarly.com/blog/5-most-effective-methods-for-avoiding-plagiarism/ (Accessed: 5 January 2021).

Chacon, S. (no date) *Git - About Version Control*. Available at: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control (Accessed: 30 November 2020).

Chapman, C. (2010) *Color Theory for Designers, Part 1: The Meaning of Color*, *Smashing Magazine*. Available at: https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/ (Accessed: 5 January 2021).

Cherry, K. (2020) *How Does the Color Green Impact Mood and Behavior?*, *Verywell Mind*. Available at: https://www.verywellmind.com/color-psychology-green-2795817 (Accessed: 4 December 2020).

Configuring SVN and Eclipse (no date). Available at: https://www.softwaretestinghelp.com/eclipse/configuring-svn-and-eclipse/ (Accessed: 30 November 2020).

CSS Tutorial (no date). Available at: https://www.w3schools.com/css/default.asp (Accessed: 30 November 2020).

DeRemer, D. E. (2015) 'Print and Web Design for Older Adults', *Discovery Eye Foundation*, 28 July. Available at: https://discoveryeye.org/print-and-web-design-for-older-adults/ (Accessed: 4 December 2020).

Frequently Answered Questions | Open Source Initiative (no date). Available at: https://opensource.org/faq#commercial (Accessed: 5 January 2021).

Gurubatham, M. (2015) 'Eclipse Window Builder Tutorial for GUI Creation | Examples Java Code Geeks - 2020', *Examples Java Code Geeks*, 24 December. Available at: https://examples.javacodegeeks.com/desktop-java/ide/eclipse/eclipse-window-builder-tutorial-gui-creation/ (Accessed: 30 November 2020).

Holmes, R. (2017) 'Writing BDD Test Scenarios', *Department of Product*, 4 July. Available at: https://www.departmentofproduct.com/blog/writing-bdd-test-scenarios/ (Accessed: 30 November 2020).

HTML Tutorial (no date). Available at: https://www.w3schools.com/html/default.asp (Accessed: 30 November 2020).

Krug, S. (2014) 'Don't make me think, revisited: a common sense approach to Web usability', *Choice Reviews Online*, 51(11), pp. 51-6218-51–6218. doi: 10.5860/CHOICE.51-6218.

Larkin, P. (2020) 'JavaFX vs Swing: The Key Differences', *Career Karma*, 7 July. Available at: https://careerkarma.com/blog/javafx-vs-java-swing/ (Accessed: 5 January 2021).

Licences and Rights (no date) *The Copyright Hub*. Available at: http://www.copyrighthub.org/copyright-resources/licences-and-rights/ (Accessed: 5 January 2021).

Lockett, E. (2020) *What Do Colorblind People See? Types, Images, and More*, *Healthline*. Available at: https://www.healthline.com/health/eye-health/what-do-colorblind-people-see (Accessed: 10 January 2021).

Mendonca, P. (2020) 'Human-Computer Interaction: What Is It And Why It Matters', *Exaud*, 12 May. Available at: https://exaud.com/human-computer-interaction/ (Accessed: 1 December 2020).

Morris, B. (2013) *10 Colors That Increase Sales, and Why*, *Business 2 Community*. Available at: https://www.business2community.com/marketing/10-colors-that-increase-sales-and-why-0366997 (Accessed: 9 January 2021).

Muniz, F. (2016) *An Introduction To Heuristic Evaluation*, *Usability Geek*. Available at: https://usabilitygeek.com/heuristic-evaluation-introduction/ (Accessed: 18 November 2020).

Nielsen, J. (1994) *10 Usability Heuristics for User Interface Design*, *Nielsen Norman Group*. Available at: https://www.nngroup.com/articles/ten-usability-heuristics/ (Accessed: 18 November 2020).

Peppers, J. (2020) *jonathanpeppers/Xamarin.Forms.Mocks*. Available at: https://github.com/jonathanpeppers/Xamarin.Forms.Mocks (Accessed: 30 November 2020).

Renk Etkisi | The Effect of Color | Color Preferences in the Elderly (no date). Available at: http://renketkisi.com/en/color-preferences-in-the-elderly.html (Accessed: 2 December 2020).

Ross (2010) 'HCI -Human computer Interaction: COLOUR THEORY IN BUSINESS/WEB MARKETING & DESIGN', *HCI -Human computer Interaction*, 5 June. Available at: http://hci-

humancomputerinteraction.blogspot.com/2010/06/colour-theory-in-businessweb-marketing.html (Accessed: 4 December 2020).

Rouse, M. (no date) *What is waterfall model? - Definition from WhatIs.com, SearchSoftwareQuality*. Available at: https://searchsoftwarequality.techtarget.com/definition/waterfall-model (Accessed: 30 November 2020).

Singh, C. (2017) 'How to configure Apache Tomcat Server in Eclipse IDE', *beginnersbook.com*, 23 June. Available at: https://beginnersbook.com/2017/06/how-to-configure-apache-tomcat-server-in-eclipse-ide/ (Accessed: 30 November 2020).

'The Best Colors For Productivity And Creativity In Your Workplace' (no date) *Digital Synopsis*. Available at: https://digitalsynopsis.com/design/workplace-office-productive-color-palette-psychology/ (Accessed: 4 December 2020).

The Color Gray (no date) *Empowered By Color*. Available at: https://www.empower-yourself-with-color-psychology.com/color-gray.html (Accessed: 10 January 2021).

VisualSVN | Getting Started (no date). Available at: https://www.visualsvn.com/visualsvn/getting-started/ (Accessed: 30 November 2020).

What is Color Theory? (no date) *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/topics/color-theory (Accessed: 6 December 2020).

'What is Human-Computer Interaction (HCI)' (2019) *GetSmarter Blog*, 15 February. Available at: https://www.getsmarter.com/blog/career-advice/what-is-human-computer-interaction-hci/ (Accessed: 18 November 2020).

What is Xamarin? | .NET (no date) *Microsoft*. Available at: https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin (Accessed: 5 January 2021).

Whitenton, K. (2013) *Minimize Cognitive Load to Maximize Usability, Nielsen Norman Group*. Available at: https://www.nngroup.com/articles/minimize-cognitive-load/ (Accessed: 5 January 2021).

Xamarin Tutorial | Hello World in 10 minutes | .NET (no date) *Microsoft*. Available at: https://dotnet.microsoft.com/learn/xamarin/hello-world-tutorial/intro (Accessed: 30 November 2020).

Zia, M. M. (2016) 'Eclipse Web Development Tutorial | Examples Java Code Geeks - 2020', *Examples Java Code Geeks*, 4 October. Available at: https://examples.javacodegeeks.com/desktop-java/ide/eclipse/eclipse-web-development-tutorial/ (Accessed: 30 November 2020).

Appendix

Projection Specification

Project outline: A study in (HCI) human computer interaction
Aims: To compare various user interfaces and evaluate their design in terms of human usability

Background: User interfaces are becoming increasingly more important as the world conducts a web- based conversation with itself, along with the continuing computerization of products and facilities. When interfaces are situated in safety-critical contexts, their design and usability can be a matter of life and death: consider the fatalities associated with the Therac-25 radiation therapy machine. The USA Gore-Bush presidential campaign in 2000 was significantly disrupted by voter confusion over the computerized butterfly ballot design. Other classic interface issues include users mistaking their CD-ROM tray for a cupholder, or looking for the "any key". In terms of e-commerce, companies invest in the design of customer web-sites with consideration to visual appeal and usability. Current directions for interface applications include mobile, wearable and ubiquitous computing.

HCI issues include: colour theory; human perception; haptic/tactile technology; gender / age /cultural / special needs issues; speech recognition / generation; graphic design; cognitive issues such as memory, learning and problem solving; design of fonts; navigation; feedback to the user; usability; aesthetics; ethical issues; and interface problems.

For this project the student will design and implement at least 3 different software interfaces (just focusing on the interface) - for instance a web-page/site, a data-base, an interactive sketch tool, a distance learning facility, or a GUI. A more challenging goal is to implement a mobile interface such as for the Android operating system for touchscreen devices.

The report will comprise a comprehensive survey on HCI discussing both software and hardware interfaces. In particular, the software interfaces implemented by the student will be evaluated in the report in terms of HCI principles.

This project is not based on any of your courses, therefore some HCI material will be provided.

Early Deliverables
1. A text-based (non-interactive) monochrome web-page
2. A colourful web-site including images and navigation
3. GUI built with buttons etc.
4. Report: about 15 pages including sketches of designs.

Final Deliverables
1. Design and implement a more advanced interface(s)
2. Complete report
3. The programs must have an object-oriented design, using modern software engineering principles.
4. The report will describe the software engineering processes involved in generating your software.
5. The report will include comparisons of interfaces with a discussion of their meanings.
6. The report will include a User Manual.

Prerequisites: Ability to research a new topic and implement designs.

Installation Manual

GUI

In order to install the GUI, a jar executable file has been created. By installing this file, you are able to run the GUI I have made on any computer. This jar file can be found in the GUI folder, once in this folder you need to go to the src folder where you will find a file named GUI.jar. This is the file you will need to install.

GUI Proof Of Concept

To install the GUI proof of concept program, a jar executable file has been created. Again, by installing this file you are able to run the GUI I have made on any computer. The jar file can be found in the GUIDemo folder, once in this folder you need to go to the src folder where you will find a file named GUIDemo.jar. This is the file you need to install.

Website

The website doesn't require any installation to run the program, assuming you have a web browser to access it. You simply have to access the folder where the HTML and CSS files exist. This means once you are in the Website folder, you just need to go into the trunk folder and then from there into the WebContent folder. To then access the website, you would click the MainPage.html file. If you would like to install the file to your computer, you would have to install all of the WebContent folder on to your computer.

Website Proof Of Concept

Again, this website won't require any installation to run the program, assuming you have a web browser to access it. To access the website, you would just need to access the folder where the HTML and CSS files exist. This means once you are in the WebsiteDemo folder, you would need to go into the trunk folder and then into the WebContent folder and click the MainPage.html file to access the website. If you would like to install the file to your computer, you would have to install all of the WebContent folder on to your computer.

App

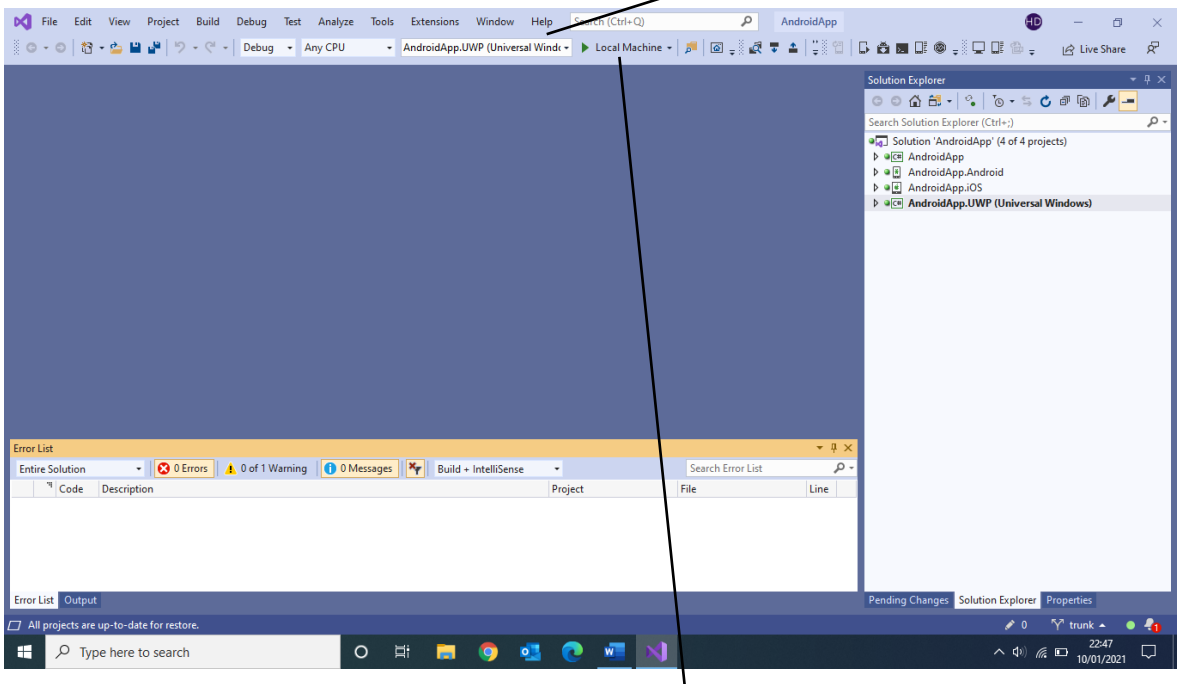
The app can be run on both Android and Windows (UWP). To run the program, you would have to install Visual Studios. Then you need to open the solution on Visual Studios using the AndroidApp.sln file found in the trunk folder which will be found in the AndroidApp folder. Then there will be options available at the top of your screen on Visual Studios via a dropdown, you can either run it on windows (UWP) or android (see Figure 29 label regarding running via windows or android). In order to run it on windows you would simply press the Local Machine button near the dropdown once windows has been selected on it, this will load the app and also install the app on your computer. To run this on android will however require more effort, as once you have selected the android option on the dropdown you will either have to download an emulator of an android device which visual studios will walk you through how to download or connect your own android device to your computer which should be picked up by Visual Studios. Then you will press the button near the dropdown to download the app onto your device which should show either your android device connected or an emulator (you can pick your choice here as shown in Figure 30). Once you press the button the app will load up on the screen and install on your device. For more help refer to Figure 29 and 30 below.

App Proof Of Concept

The proof of concept app program can be run on Android and Windows (UWP). Again, to run the program you would have to install Visual Studios. Then you need to open the solution on Visual Studios using the DemoApp.sln file found in the trunk folder which will be found in the AppDemo folder. Then there will be options available at the top of your screen on Visual Studios via a dropdown, you can either run it on windows (UWP) or android (see Figure 29 label regarding running via windows or android). In order to run it on windows you would simply press the Local Machine button near the dropdown once windows has been selected, this will load the app and also install the app on your computer. To run this on android will however require more effort, as once you have selected the android option on the dropdown you will either have to download an emulator of an android device which visual studios will walk you through how to download or connect your own android device to your computer which should be picked up by Visual Studios. Then you will press the button near the dropdown which will either show your android device connected or an emulator (you can pick your choice here as shown in Figure 30) to download the app onto your device. Once you press the button the app will load up on the screen and install on your device. For more help refer to Figure 29 and 30.

Dropdown used to select if you want to load app via windows or android.

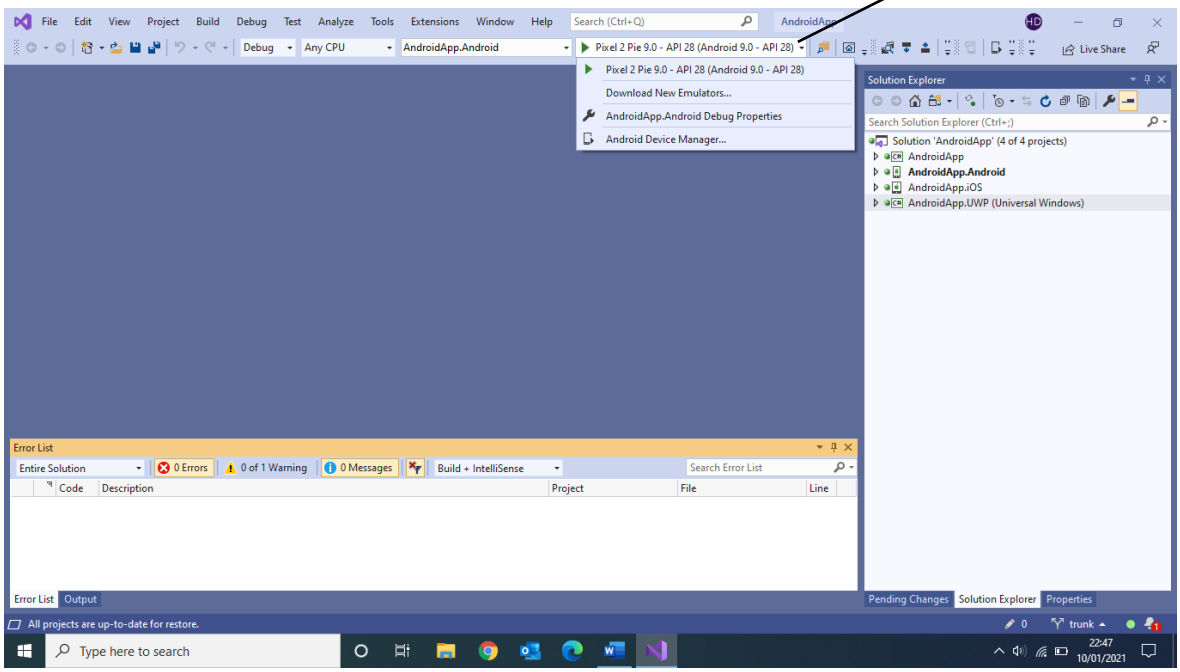
Figure 29. Figure shows how to run app via windows (UWP)



Pressing ‘Local Machine’ will then load the app and install onto your computer.

When running via android, if you decide to run app via android device it will appear in this dropdown. If you run via emulator, the emulator should automatically appear as the first choice once Android is selected from the other dropdown.

Figure 30. Figure shows how to run app via android (emulator or android device)



Subdirectory Structure

The subdirectory structure of my submission directory is fairly straightforward. All of my work done for this project is located in the 'halfunitproject' folder. In this folder consists of folders of my three interfaces I have created and folders of my three proof of concept programs created. There also exists a folder named Documents which consists of my BDD tests done for my three interfaces and my final report. Finally, there exists a README.txt file in the 'halfunitproject' folder which describes the structure to my directory as well in more detail. All of my interfaces and proof of concept programs share a similar format of files inside them, they all consist of a trunk, branches and tags. The trunk consists of the main source code for the program, the branch folder was used to create a number of branches to implement features/changes to the trunk without affecting it. The tags folder consists of versions of the main files/code (trunk).