Final Year Project Report

Full Unit – Interim Report

A study in (HCI) Human Computer Interaction

Lewis Edmund

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: Sara Bernardini

Department of Computer Science

Royal Holloway, University of London

December 11, 2019

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 Where other published and unpublished source materials have been used, these have been acknowledged.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: 4928

Student Name: Lewis Edmund

Date of Submission: December 11th 2019

Signature: Lewis Edmund

Table of Contents

Project Specification

For this project the student will design and implement at least 3 different software interfaces (just focussing 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.

Abstract

The reason why I chose to do this project is because I am highly interested in the concept of HCI and I wish to pursue this field post-university in relation to my career aspirations. Whilst studying for my A-levels, I had the opportunity to participate in a work experience placement with Sky. During the week I worked with a group of front-end developers on a streaming application tailored for business clients. This hands-on experience in the agile process of front-end development encouraged me to think more about how humans interact with computers and the fine details which go into designing and developing a usable interface. In addition, I started to comprehend the complexity of a usable interface during my second year team project, which was based on creating a website and allowed me to explore and design a user interface. After completing the team project and gaining further insight into how users want interfaces to work and look, I have decided to conduct my individual project within this field of study.

Furthermore, within this project, I would like to incorporate one of my passions; sport. Using sport as a focal point, this will provide a solid foundation for design within the interfaces I intend to produce. Moreover, my project is focusing on sport because having participated in a university sports team over the past few years and being elected as captain, I understand the difficulty of managing fixtures, coaches and players. Also, I empathize with students who partake in any sport at university, as they do not have a centralised point to access the information they may require. Therefore, I will be designing interfaces to be used by students and coaches in order to check fixtures, post results and view the availability of students for fixtures in each team, while also providing further information to recreational students about memberships, opening dates and times for sports venues and contact details.

Overall, the aim of my project is to explore the accessibility options available when developing interfaces and how these features affect human interaction. In order to do this, I will develop user interfaces in three formats; text-based, web-based, mobile application. As a result, I will be able to compare how HCI can be designed at different levels of interface complexity.

Chapter 1: **Background Theory**

Human-Computer Interaction (HCI) research began in the early 1980s due to the rise of personal computing in the late 1970s. Prior to the 1980s, computing was carried out by IT professionals or dedicated individuals as a hobby. As a result, Carrol [1] states that the computer science community needed to find a way to present information for a wide variety of people. HCI theory is broken up into three components; how humans process information, how computers store and present information to its users and the interaction that occurs between the two, in order to address the required action. I will now discuss these components further and explain how they come together to have an impact on interface design, whilst highlighting some accessibility issues that affect how people interact with computers. The following sequence of definitions is based on [2], with simplifications due to the fact we are only considering finite widgets.

1.1 Human

Firstly, Humans receive information through certain channels, such as; visual, auditory, haptic and movement. Information is then stored in classes of memory, for example; sensory, short-term, long-term. It is then processed via reasoning, problem-solving, skill acquisition and error known as cognitive tasks. In addition, another process that falls under the umbrella of cognition is attention. This is the procedure of selecting things to concentrate on at one point in time from a range of possibilities, allowing us to focus on the information that is relevant to the task we want to achieve at that moment in time. The success of the task is based on whether we have clear goals and how easy the environmental information is to interpret. As a result, understanding human attention is paramount when dealing with HCI.

As previously mentioned, the classes of memory that humans use to store information is then used to respond appropriately to certain tasks. However, it is impossible for humans to remember everything we store in memory, otherwise, our brains would overload. Rather, information is filtered based on how much attention is paid to it, as it is known that humans tend to pay more attention to colours and shapes compared to numbers, words or speech. Therefore, it is evident that parts of cognition come together in order for the information to be processed in our brains. Hence, when dealing with HCI, it is important to reduce the load on the user's memory by using simple procedures and prioritising recognition using menus and icons.

The human ability to learn is another important aspect to consider when discussing HCI theory which, again uses the concepts of attention and memory, to recall similar past experiences and use previous responses to act upon a required task. Humans tend to learn a lot more from doing, rather than following a set of instructions, and as a result, it is important to design with exploration in mind, in order to allow humans to learn the most efficient and productive solutions to problems. Furthermore, learning is also about improving upon mistakes and enabling users to return to previous states to encourage further development and helps guide them through the tasks. Overall, as humans we process information in a variety of ways and using the concepts of cognition, we can employ the use of computers to deal with complex tasks much faster and easier than we could on our own.

1.2 Computer

Secondly, Computers have similar human features, for example, software packages are used to process information with the help of input/output devices. Also, computers have other human similarities, such as; memory and processing power, which make up the hardware of a computer.

However, computers have the ability to perform these tasks at a faster rate, providing solutions to complex queries, that couldn't be solved by the average human in the same time frame.

Additionally, computing has evolved over time by increasing its accessibility in order to meet the requirements of users. In the history of accessibility, a substantial number of inventions were created to assist people in completing tasks, for example; in 1808 the first typewriter was built in order to help a blind person write legibly. These historical developments directly correlate to how a variety of users, with different physical and mental conditions, are able to perform certain tasks and use computers. There are a number of assistive technologies that have been produced to support HCI, both in hardware and software. Firstly, an example of this in software is; screen readers. This application was developed by IBM in 1986 to support visually impaired staff, by reading aloud content on computer screens and quickly became available to all users of personal computers. Secondly, an example of assistive technology in hardware is an adaptive switch. This allows people with movement-limiting disabilities to use computers without the need to perform complex actions, by offering easier movement solutions, such as; pressing a button. There are different types of switches available, such as; joysticks, buttons, and sound, which can all be configured to a user's specific needs. The use of assistive technology, therefore, allows computers to adapt to different user bases and environments making user interaction simple for all.

Earlier on, I commented on the background theory, highlighting the rise of personal computing is the primary source of HCI research. At this point in the short lifespan of computing, industry professionals were referred to as; operators, who used computers to perform specific tasks relating to one end product, such as; producing medical statistics. Computers started to become programmable by trained users due to the developments in hardware, which in turn made the machines smaller, cheaper, faster and easier to use.

In 1977, there was a major breakthrough for personal computing. Jobs and Wozniak exhibited the; Apple 2, which had a built-in programming language called; Beginners All-purpose Symbolic Instruction Code (BASIC). This language was developed with a design philosophy emphasizing the "ease of use". It was also created to be supported by the hardware, which can store programs and data on a compressed storage device. As a result, this became the root cause of computer production, whereby, non-professional users could operate these machines without being subjected to intricate commands and system dialogs. Hence, there is now a new community of users that have different ways of thinking compared to industry professionals, which is when Carroll [1] believes the notion of cognitive engineering arose to talk about applications that are informed systematically and scientifically.

1.3 Interaction

Thirdly, Cognitive Engineering defines the interaction between a human and a computer, in order to comprehend the translations between what the user wants and what the system does. This started HCI research, by developing techniques to evaluate how humans interacted with computers, and sequentially, relevant documentation was produced to analyse how tasks were performed. The documentation outputted by systems used in personal computing then moved away from producing technical descriptions and prioritised supporting users to achieve goals by recognising and recovering from the error. This supports the process of learning, as I previously mentioned the interaction should be designed with a way to guide users through error, making system-based tasks easy to learn and easy to achieve.

To influence these interactions, interfaces are used, thus, the style of an interface is determined by the style of interaction that needs to occur. At the origin of personal computing, HCI theory was limited to desktop applications, such as; spreadsheets and word-processing tasks. Since then, desktop interfaces have gone through many changes, starting with files and folders being represented as icons. However, over time the user's desktops became cluttered with icons,

therefore, a search functionality was incorporated to graphical user interfaces, in order to reduce the need to keep icons visible on the desktop. Furthermore, the rise of the internet drove HCI further away from the desktop through applications, such as; email. This permitted people to start using computers to interact with other users in this rapidly developing environment, which became known as; social computing. These interfaces concentrated on collaborative work through; instant messaging, forums and online communities, such as; Facebook and GitHub. Also, another reason why interfaces are evolving is because of the introduction of different computing devices. Nowadays, computing has been incorporated into various parts of daily tasks for humans, most notably; laptops, mobile phones, and cars. As a result, desktop computing has come a long way since its inception, where it now informs interface design to enhance human activity and experience with various systems.

Another significant aspect of interface design is Usability Testing, Bastien [3] explains how it is used to inform software development, whereby, non-industry users are observed in order to identify where they encounter problems and experience confusion. This is an important part of the interaction design, as it identifies non-functional issues with interfaces. These tests measure user interfaces (UI's) based on how easy it is for a user to reach their goals. The bias of industry professionals is completely removed, and the focus is put on the end users, who will have different skills and experience to developers and testers.

1.4 Accessibility Issues

Fourthly, though users share common capabilities, they also have distinct differences to be considered when designing HCI concepts. There are several HCI accessibility issues that user interface designers, in many countries, are legally obliged to incorporate. There is a common misconception that these issues are purely related to some form of physical or mental disability, however, it can be any form of restriction that a user may have in being able to productively use an interface, such as; having a slow network connection.

Following on from that, I will now discuss several areas that should be looked at when designing with accessibility in mind. Foremost, it is important to note that there are different forms of visual issues that need to be catered for. For example, users can have various deficiencies in colour vision which can include perceiving colour contrasts differently than in normal sight, or the inability to see certain colours or even any colour and extreme sensitivity to flickering lights, also known as; photosensitive epilepsy. Consequently, it is vital to consider users that have limited or no vision at all. These cases vary from; tunnel vision, where users can only see central elements or; blurred vision, where text becomes very difficult to read. Therefore, in each of these cases there are some practices which can be employed, such as; ensuring content is separate from the structure of the interface. This allows web browsers to interpret the information and present it in alternative ways to support the user's requirements, for example; providing text substitutes for images allows screen readers to describe them. Additionally, avoiding the use of colours and instead using textures and icons, to differentiate between elements and refrain from harsh flashes or transition animations.

Furthermore, Interfaces also cater for mobility issues, which for web based products can be difficult to design for. It is, therefore, important that the environment in which the interface is being used in, is taken into consideration, as this may affect users which will have difficulties accessing or moving around. Thus, it is vital to allow alternative input devices to be used, such as; eye tracking applications and mouth sticks, in order to aid those without motor function in certain areas. As well as, ensuring that precise mouse positioning is not required, and allowing keyboards to be used to traverse through links on browsers. In addition, designing web interfaces for those with auditory disabilities are also tricky, unless, the product has a multimedia aspect to it. In these cases, where a UI requires the use of audio clips or videos, transcripts should be provided and made available through subtitles. This can help those with auditory concerns, but may also benefit non-native speakers of the language, used in said media.

Additionally, cognitive disabilities will also affect the accessibility of a UI in different ways. For example; users can have issues with spatial reasoning, which affects their ability to visualise the structure of information presented to them, however, implementing a sitemap alongside simple and efficient navigation, will aid this issue. Alternatively, some users have problems reading large amounts of text, therefore, it is wise to promote scanning of key words and links to help ease the use of the interface. On top of that, users misspell words all the time, regardless of their cognitive ability. Thus, providing a spell checker or similarity search, when requiring user text input, is a great design idea which will benefit users from other languages as well.

Overall, exploring the background theory of HCI has enabled me to comprehend its history and how user accessibility has evolved alongside it. It became evident to me that the turning point for HCI research was the rise of personal computing. The creation of this global community, where the world of computers welcomed a host of new users, created a vast collection of cognitive thinking skills which now need to be considered. For example; designing with usability in mind, is a key concept of HCI. Hence being able to produce intuitive and accessible interfaces is essential to enhance user experiences and promote inclusivity.

1.5 Applied Knowledge

Moving on from that, I will now discuss a couple of the guidelines that are used in the real world to define accessibility attributes for UI's. One guideline is; Accessible Rich Internet Applications (ARIA). This is a specification used for web UI's, which allows developers to implement solutions to accessibility issues that base HTML cannot solve on its own. Elements of interfaces can have specified attributes which provide extra information or checks when it is being translated into accessible formats for users. In the following example, the Google developer website [4] explains a use of ARIA for users with sight issues:

```
<li tabindex=”0” class= “checkbox” role=”checkbox”
    checked aria-checked=”true”>
            Receive promotional offers
</li>
```

This code snippet uses the ARIA attributes “role” and “aria-checked” to explicitly identify the element as a checkbox, with a Boolean check, we are verifying its use in the accessibility tree. It will now allow a screen reader to correctly report it as a checkbox.

In addition to this, the Web Content Accessibility Guidelines (WCAG) are another set of standards. They are essentially a step by step manual on making websites or application interfaces, which are usable to all users. WCAG are categorised into four principles [5]. 1) Perceivable; “users must be able to comprehend the information being depicted. It can't be invisible to all their senses”. 2) Operable; “the interface cannot require interaction that a user cannot perform”. 3) Understandable; “users must be able to understand the information as well as the operation of the user interface”. 4) Robust; “Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies, as technologies and user agents evolve, the content should remain accessible”. I will use the guidelines within these principles to assist in the development of the web UI and app UI.

Chapter 2: **Summary of Completed Work**

For my web-based UI, I started reviewing what I had learnt from my team project in year 2. Within this project, I was part of the front end team and responsible for the design and implementation of the e-commerce interface that the customers would use. I began by thinking about what pages I would like to present, how they would interact with each other and how they would look. Starting with the foundation of any website, I constructed a standard home page using HTML, and gave it a title heading and some text to test that the structure of the file was correct. From this point on, my development strategy was to put myself in the mind of someone using the website and think about what tasks I would like to achieve. With this in mind, I created some links to various pages I would expect to be used for a sports website, for which I had to create individual page files for each. Then, when navigating to one of these pages, I needed a way to get back to the home page or somewhere else on the website, hence, the implementation for a navigation bar was needed for all pages. I defined a div for the nav bar and applied some basic stylings to clearly define it to users.

Next, I went across each of the pages and added some relevant content. Firstly, I started with the fixtures page and I used a table to list some upcoming fixtures for different clubs within the university. This was later changed to also include recent results and as a result, it became a fixture centre page to allow users to keep updated on various clubs. Moving on from that, I then went on to focus on the individual clubs pages, where I wanted to use images to link to the individual pages for each club. In order to make that possible, I carried out some research to find some copyright free images and I made notes on any images that required acknowledgement. Additionally, I used a card style format to layout the images, for each of the eight example clubs, and applied some CSS to clean up their presentation. On top of that, rather than creating similar pages for all clubs, I used the American Football page as an example of how they would all look. For example, by filling content with headings and tables with relevant information on the clubs, in terms of training, fixtures and-so-forth.

Following on from that, I looked at a login feature to be incorporated into the navigation bar of each page, in order to allow users to be able to access their accounts from anywhere on the site. My first thought was to make it simple for users to login via the nav bar, using a form directly inside the div. But having changed my mind, I decided to make it as a simple login link, identical to the other links on the nav bar, with some CSS to place it on the right side of the page. By creating this link, it meant I needed a login page with no navigation just a single div which has a title and the login form, with a submit button and a cancel button which returns users to the home page.

Moving on, the next pages I looked at were the venue and membership pages, which also followed the card style format used in other pages. The venue page has a set of three cards with; titles, images and descriptions of some locations where sport is hosted. Whereas, the membership page uses cards for each unique membership type. There are two divs one for student opportunities membership and the other for gym membership. Users can add each membership card to a basket and checkout features will be implemented next term.

For the text-based UI, I began by conducting some external research into what this meant. From my findings, I realised that this was similar to Teletext, which I remember using as a child to check Football results, news and league standings. Once collecting the information from my research, this gave me a target of what I needed to achieve. Firstly, I looked into Java libraries that supported text-based UI design and I came across Lanterna. This library allows users to develop and run UI's in text only environment, straight out of an IDE. Using Eclipse, I created a Maven project and added relevant .jar files for Junit and Lanterna. I then followed a couple of tutorials on how to use the library and this aided me to develop a foundation for the interface, as I had a good idea of how it would work. This interface is made up of a terminal, which runs until the escape key is pressed and other keystrokes are used to navigate through different pages of the interface.

At this stage, I had the following pages that I could navigate to by using different keystrokes; home, clubs, fixtures and results, venues, membership. Following on from that, I then added content to each page using features of the library to draw boxes for tables etc.

Chapter 3: **Planning & Time-Scale**

For the remainder of the project, I will be designing and developing an Android application with touchscreen capabilities, using the work that I have completed thus far. Firstly, I will need to research into Android app development, to collect information on the technologies that I will require and any tutorials that may be of use to me, during the development process. Leading on from this, I will need to sketch some design ideas, using the web-based UI as a point of reference. Following on from that, this application will provide users with more functionality compared to the web-based UI, hence, an extensive UML class diagram will be needed. The UML will depict the structural and behavioural features of the interface in terms of; what various parts of the UI will do and how they will react to other parts.

The next stage of the project will be the commencement of the app development. This will require TDD, with extensive use of Junit and Javadoc to test and communicate my development journey. As a result, I will create another project with relevant dependencies inside a suitable IDE. Also, as well as unit testing, I will also need to continuously run the application on an Android device to review it. Whilst in development, I will be making regular commits on feature branches and keeping track of my work via diary entries. Overall, I aim to have a fully functioning Android application relating to university sport with various functionalities such as; logins, membership payment, posting fixture results, account management.

Moving on from that, I will also be completing my final report over term two which will build on my interim report and further discuss my final project. For this, I will extend my background research to analyse existing systems that deal with HCI in a literature review, whilst performing a critical analysis of the project's achievements and possible future enhancements. In addition to this, I will discuss technical decisions that I will make during the project and how they are beneficial or how they could be improved upon. And lastly, I will have a section on professional issues, where I will discuss the ethical and social impact of technology, regarding issues I have faced in my project.

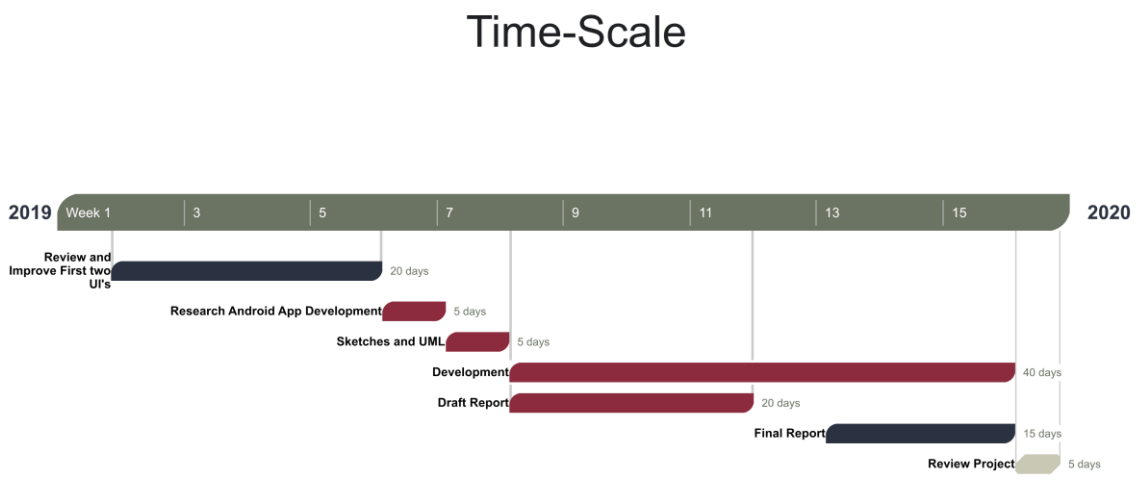


Figure 1. Time-Scale for remainder of project

Chapter 4: **Software Engineering**

I will now discuss how I have used software engineering tools and techniques to carry out the work in my project thus far:

I have made good use of my Git repository, with regular and meaningful commits made over the five-week period I've spent in development. Every commit came with a message that clearly described my thought process and the actions that I took. From my commits, I was then able to summarise each commit into a diary entry which kept track of my progress throughout the project. I also made use of branches to separate work from the different parts of my project. From my master branch, I created 3 new branches; WebUI_DEV, monochromeUI_DEV and report_writing. I could then concurrently work on these distinct parts of the project and any time a significant task or part of each branch was complete, I could merge it with the master branch and make sure it was up to date with all aspects of the project. In addition, another aspect of git that I used, is the ability to checkout a set of working files or folders. I typically used this to checkout small subtrees of my directory, to reduce conflicting files. For example, when writing my report, I would just checkout the report file and changes made would only affect that single file.

For the web-based interface, I have implemented a navigation bar that takes users from page to page throughout the website. In order to achieve this, I defined a separate div class and named it 'topnav'. I then added a list of links to each of the main pages; Home, Sports, Fixtures and Results, Venues, Membership. After this, I created a nested div class called: 'loginButton', inside the 'topnav' div and this has a single link and takes users to the login page. In addition, I used CSS to format the navigation bar in order for the main page links to line up on the left of the bar and the login link to be fixed to the right of the page (See *Figure* 2 and 3).

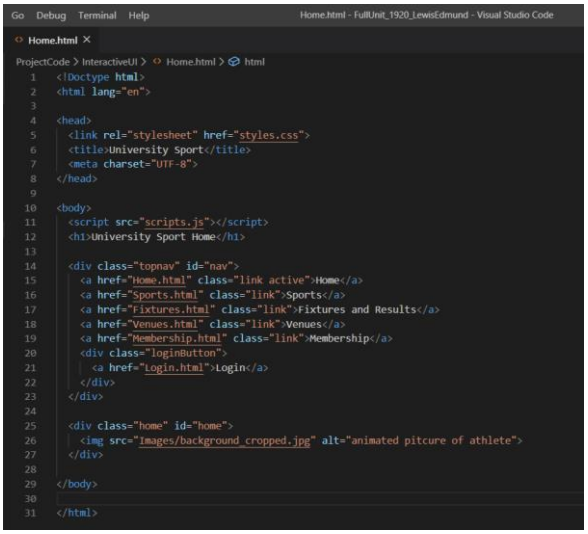
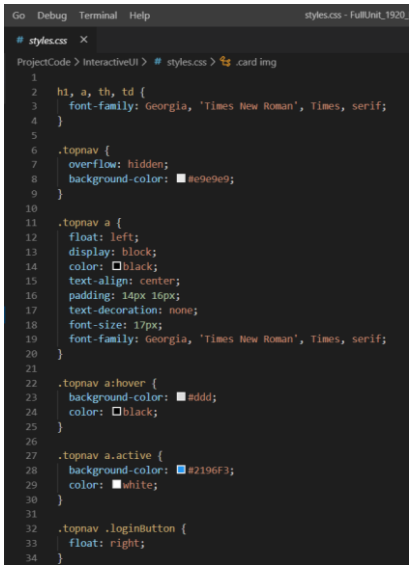


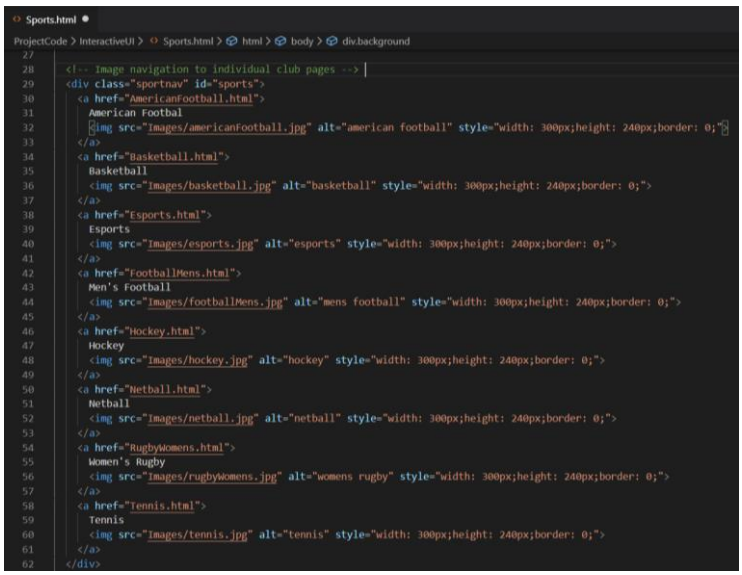
Figure 2. Home page showing navigation bar



```
Go  Debug  Terminal  Help                                   styles.css - FullUnit_1920_

# styles.css  ×

ProjectCode > InteractiveUI > # styles.css > .card img

1
2   h1, a, th, td {
3     font-family: Georgia, 'Times New Roman', Times, serif;
4   }
5
6   .topnav {
7     overflow: hidden;
8     background-color: #e9e9e9;
9   }
10
11  .topnav a {
12    float: left;
13    display: block;
14    color: black;
15    text-align: center;
16    padding: 14px 16px;
17    text-decoration: none;
18    font-size: 17px;
19    font-family: Georgia, 'Times New Roman', Times, serif;
20  }
21
22  .topnav a:hover {
23    background-color: #ddd;
24    color: black;
25  }
26
27  .topnav a.active {
28    background-color: #2196F3;
29    color: white;
30  }
31
32  .topnav .loginButton {
33    float: right;
34  }
```

Figure 3. CSS styles file showing navigation bar styling

Another technique that I have used to develop this interface is; image navigation. Similarly to the navigation bar, I am using links but also incorporating images to those links. This is achieved by nesting an image into the link content, which allows users to click any part of the image and text title, to take them to that page. I have also added alternative text to deal with some accessibility issues (See *Figure* 4).



```
Sports.html ●

ProjectCode > InteractiveUI > Sports.html > html > body > div.background

27
28  <!-- Image navigation to individual club pages -->
29  <div class="sportnav" id="sports">
30    <a href="AmericanFootball.html">
31      American Footbal
32      <img src="Images/americanFootball.jpg" alt="american football" style="width: 300px;height: 240px;border: 0;">
33    </a>
34    <a href="Basketball.html">
35      Basketball
36      <img src="Images/basketball.jpg" alt="basketball" style="width: 300px;height: 240px;border: 0;">
37    </a>
38    <a href="Esports.html">
39      Esports
40      <img src="Images/esports.jpg" alt="esports" style="width: 300px;height: 240px;border: 0;">
41    </a>
42    <a href="FootballMens.html">
43      Men's Football
44      <img src="Images/footballMens.jpg" alt="mens football" style="width: 300px;height: 240px;border: 0;">
45    </a>
46    <a href="Hockey.html">
47      Hockey
48      <img src="Images/hockey.jpg" alt="hockey" style="width: 300px;height: 240px;border: 0;">
49    </a>
50    <a href="Netball.html">
51      Netball
52      <img src="Images/netball.jpg" alt="netball" style="width: 300px;height: 240px;border: 0;">
53    </a>
54    <a href="RugbyWomens.html">
55      Women's Rugby
56      <img src="Images/rugbyWomens.jpg" alt="womens rugby" style="width: 300px;height: 240px;border: 0;">
57    </a>
58    <a href="Tennis.html">
59      Tennis
60      <img src="Images/tennis.jpg" alt="tennis" style="width: 300px;height: 240px;border: 0;">
61    </a>
62  </div>
```

Figure 4. Sports page with image navigation

Furthermore, a few pages on the website make use of tables to format information. These tables are defined by a row of header fields, followed by more rows with the information being displayed (See *Figure* 5).

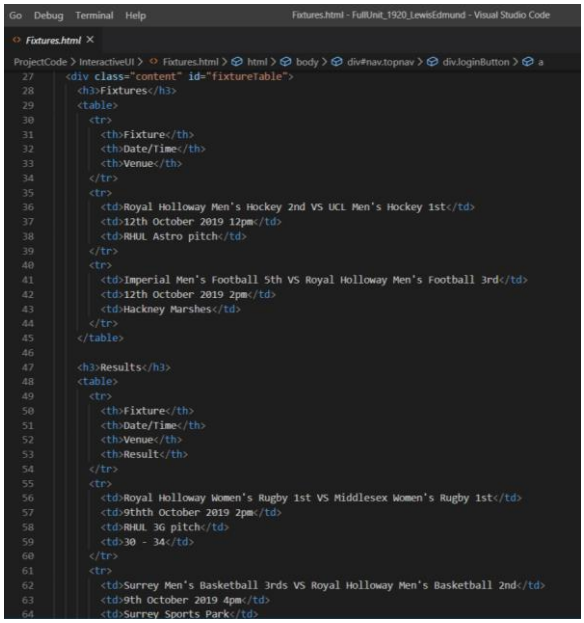


Figure 5. Fixtures page with table to present information

As well as this, the Venues page makes use of some card style elements. This feature was designed by nesting a div called; 'card', into two other nested divs; 'row' and 'column'. This allows venue information to be displayed in an organised structure (See *Figure* 6).

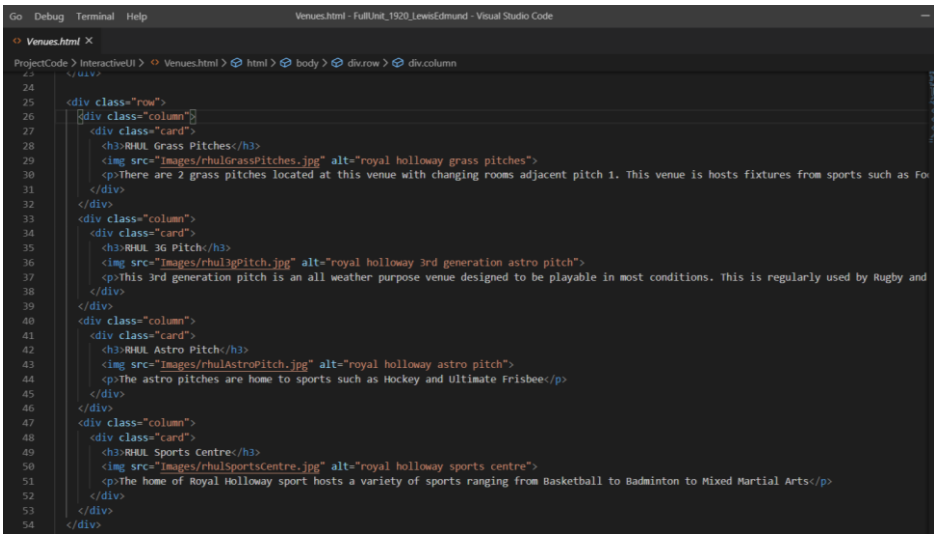


Figure 6. Venues page with nested divs to create Cards

For the text-based UI, I am making use of an external Java library; Lanterna. This UI is drawn and run from a single Main class using a terminal layer from Lanterna. The UI is set up using the variables; 'terminal' and 'textGraphics', to define a screen and displayable text. Text is then written onto the terminal using the putString method and displayed by flushing the terminal. As a result, the home page displays (See *Figure* 7).

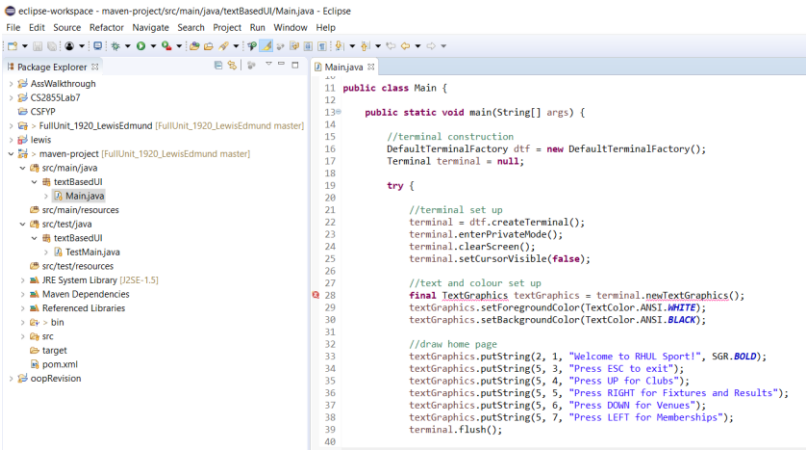


Figure 7. Home page terminal set-up

The navigation part of the UI uses keystrokes to display different pages. To keep the UI running, I have while loop with a Boolean variable; 'keepRunning', that is initialised as true. I then read the input from the terminal for keystrokes defined in a switch statement. The first case is used to close the UI, where if the ESC key is pressed, the keepRunning variable is set to false and leaves the loop, closing the screen. The next cases each display a different page based on which keystroke is read from the terminal input. For example; pressing the ArrowUp key will show the Clubs page with more options for navigation (See *Figure* 8).

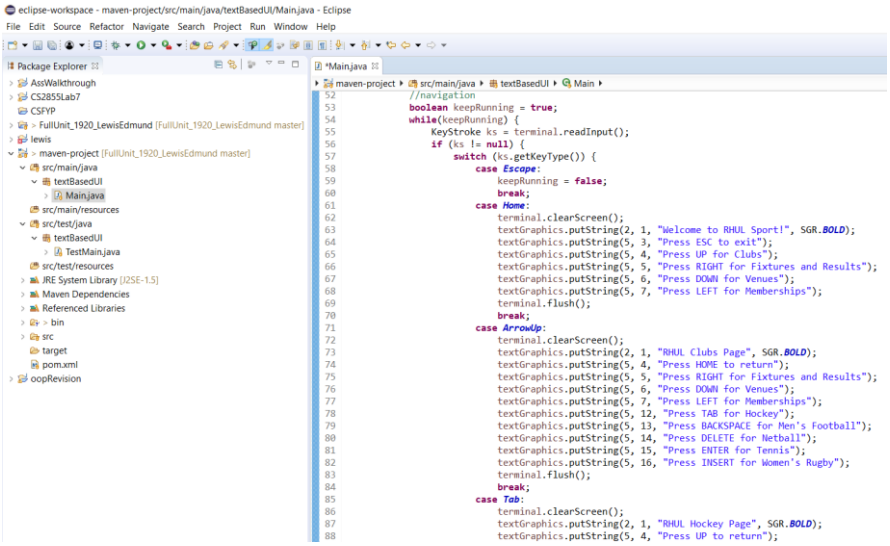


Figure 8. Navigation via keystrokes

Finally, I have some error handling code to deal with any IO exceptions that are raised. The code, which sets up the terminal and deals with input, is all enclosed in a try statement which then catches any IO exceptions and finally closes any open terminal (See *Figure* 9).

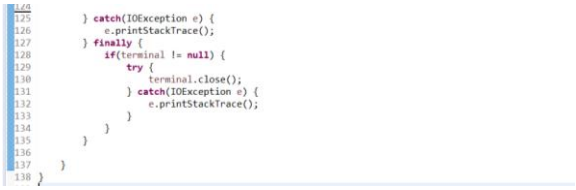


Figure 9. Error Handling

Bibliography

[1] Carroll, John M. *The Encyclopedia of Human – Computer Interaction, 2nd Ed. – Brief Intro*.

[2] Alan Dix, Janet Finlay, Gregory D.Abwod and Russell Beale. *Human-Computer Interaction, 3rd Ed. – Chapters 1-3. Pearson 2004.*

[3] J.M. Christian Bastien. *Usability testing: a review of some methodological and technical aspects of the method. Université Paul Verlaine 2008.*

[4] Meggin Kearny, Dave Gash and Alice Boxhall. *Google Web Fundamentals – Introduction to Aria*. *https://developers.google.com/web/fundamentals/accessibility/semantics-aria*

[5] Bureau of Internet Accessibility. *WCAG 2.1 A/AA Principles and Checkpoints. https://www.boia.org/wcag-2.1-a/aa-principles-and-checkpoints*

[6] Project Diary Website. *https://pd.cs.rhul.ac.uk/2019-20/*

Appendix

Diary Summary

I will now summarise the use of a diary throughout the project so far. For the first two weeks, my diary posts were kept locally on a word document due to issues with the diary website [6]. During the first week, I researched into the background theory of HCI, making notes and relevant references along the way. One of the readings that I found was about sketching user experiences, and this was proven to be a helpful tool to create some initial sketches for the UI's. The following week's tasks included; writing up some background theory, while making sure to comment sections that will need formal referencing at a later stage. Another task carried out at this stage was looking into a UML design for the UI's, this process helped me to comprehend how each of the UI's will be structured. Also, since there is little functionality in the text-based and web-based UI's, UML is not extensive, therefore, another UML will be created for the mobile app in term 2.

Following on from that, the next task I mentioned in my diary was the set-up of my GitHub repository. The decision to use Git was based on my experiences in the team project during my second year, where I learnt how to use the command line to perform Git commands and make use of the GitHub interface. Following the set-up of my Git repository, I created a Trello board with a set of tasks relating to the following; text-based UI, Web-based UI, Report. At this stage, I decided to focus on the first two UI's in term 1 and move my plan of working towards the mobile interface to term 2. The next task on my diary was about my first git commit, this commit describes how I had originally set up my directory and created a README file to define its structure.

Moving on from that, the next diary entries describe my development processes. I started by creating a branch for design towards the text-based UI, for which I had originally planned on using html to develop it. However, I soon realised that my plans for the text-based and web-based UI's were too similar. Therefore, I had a meeting with my supervisor and discussed ways to solve this. As a result, I changed my plans and decided to use Java for the text-based UI and use HTML for the web-based UI. My next diary entry comments were on the set up of the Eclipse project which will hold the source files and test cases for the text UI.

Moreover, I then go on to describe the start of development for the web-based UI. This started with a new git branch and an initial commit with a home page with a title, some text and a couple of links. Next, I spoke about the status of my report, for which I had finished the background theory section and started discussing work that had been completed thus far. Sequentially, I returned to development for the web UI and mentioned the implementation of pages for the website, with relevant content. I also highlighted the use of styles to improve the presentation of the website, and images for which some acknowledgements were noted. Additionally, I hinted at some ideas to improve the UI over Christmas, such as; checkout features and some small login functionality.

Finally, my last few diary entries were in relation to the text UI that I had changed to make use of Java. From my research into Java libraries, I had found Lanterna, which allows for UI design in text only environments. To use this library, I created a new Maven project in Eclipse and added the dependencies for Junit and Lanterna. This meant that I needed to update my directory and remove the old eclipse project and other irrelevant files. Using some Lanterna tutorials, I implemented a basic interface using a terminal layer to present text. With the use of keystrokes, I could then display different text depending on which keys were pressed. Lastly, I spoke about how I informed my supervisor about my deadline extension approval.

Program Screenshots

Text-Based UI

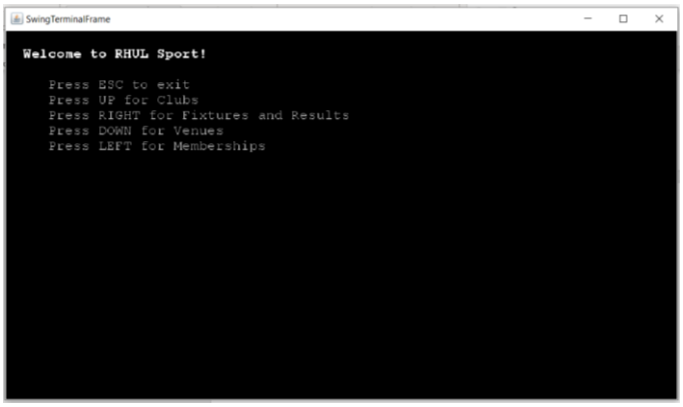


Figure 10. Home page

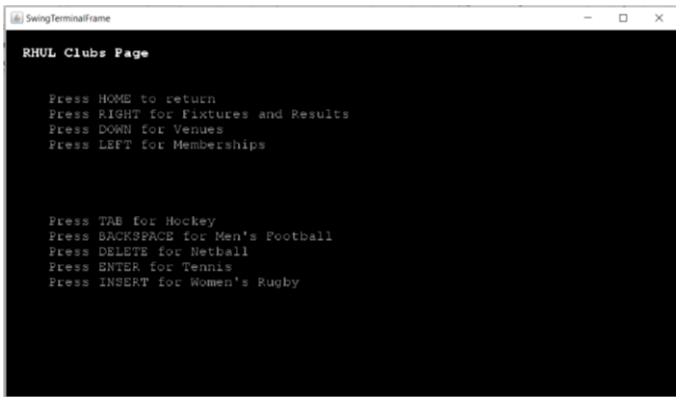


Figure 11. Clubs page

Figure 12. Hockey page

Web-Based UI

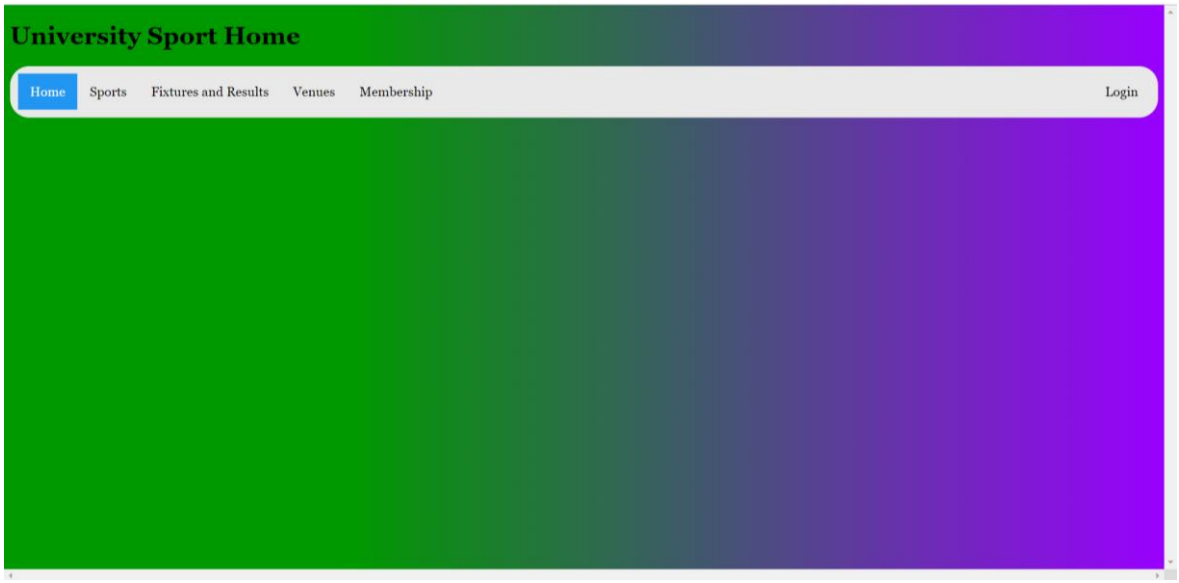


Figure 13. Home page

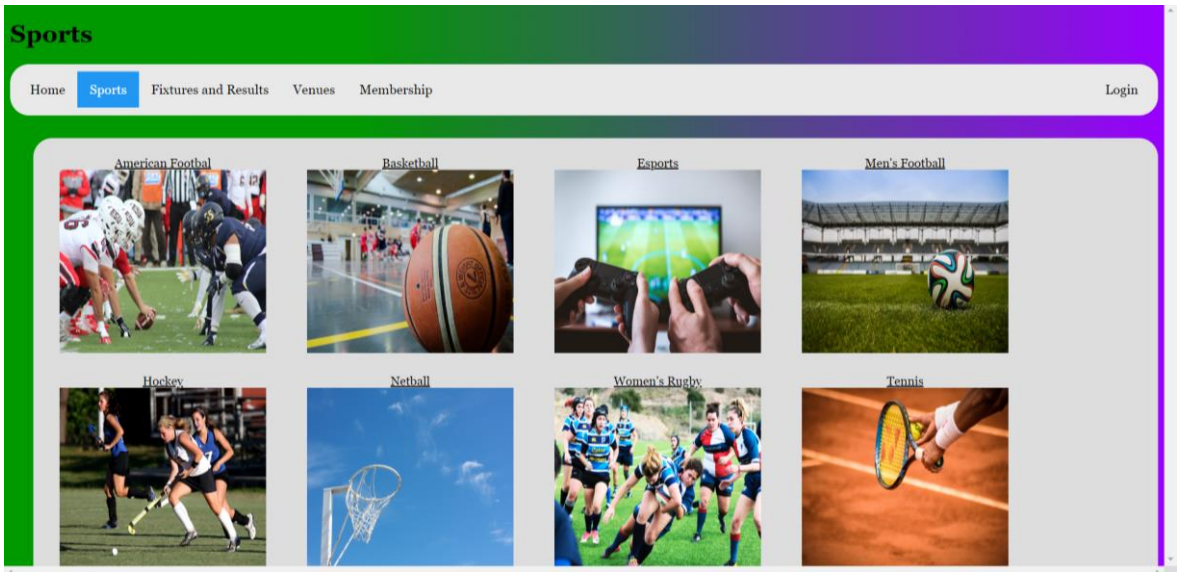


Figure 14. Sports page

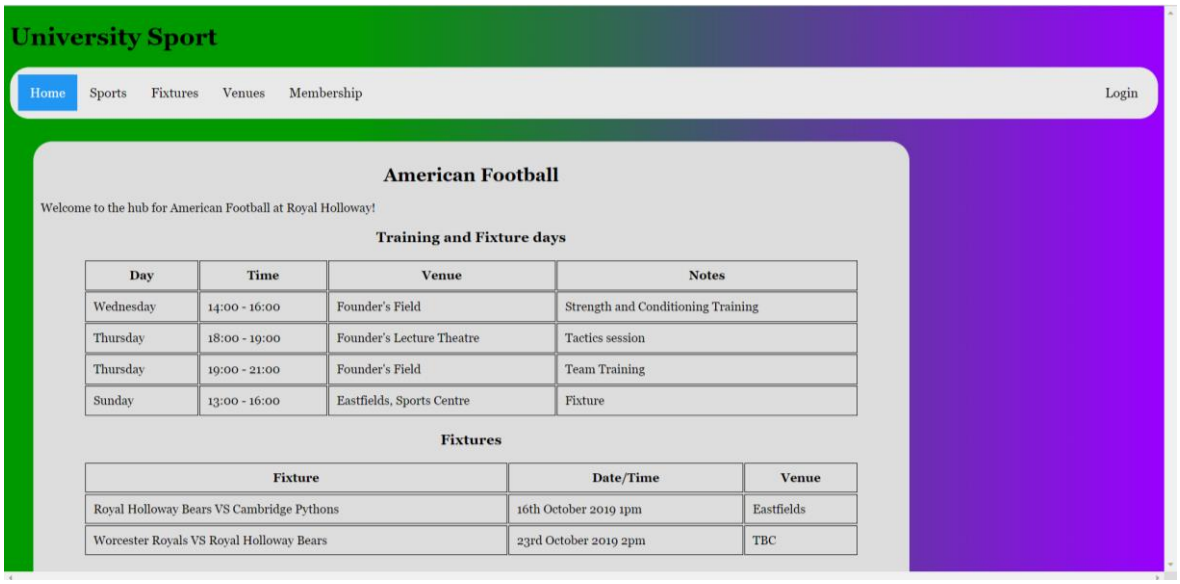


Figure 15. American Football page

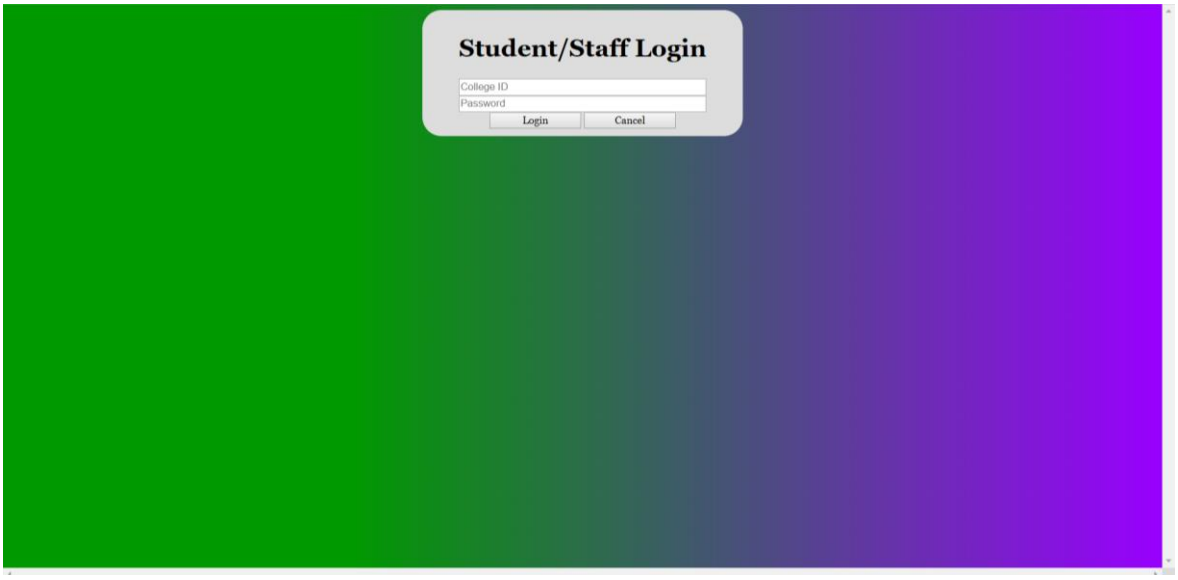


Figure 16. Login page

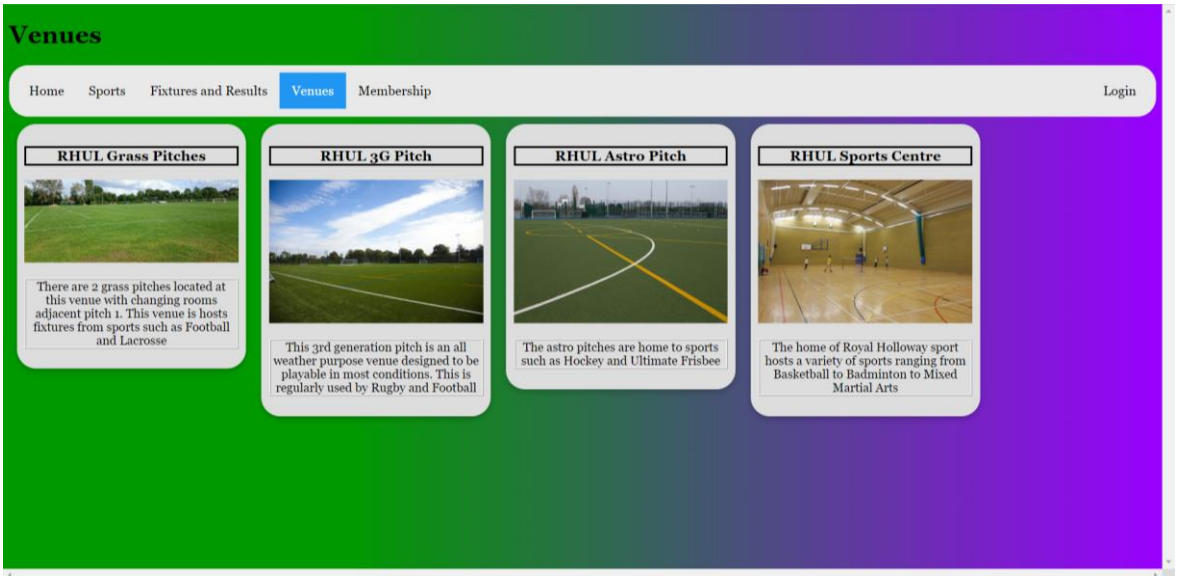


Figure 17. Venues page

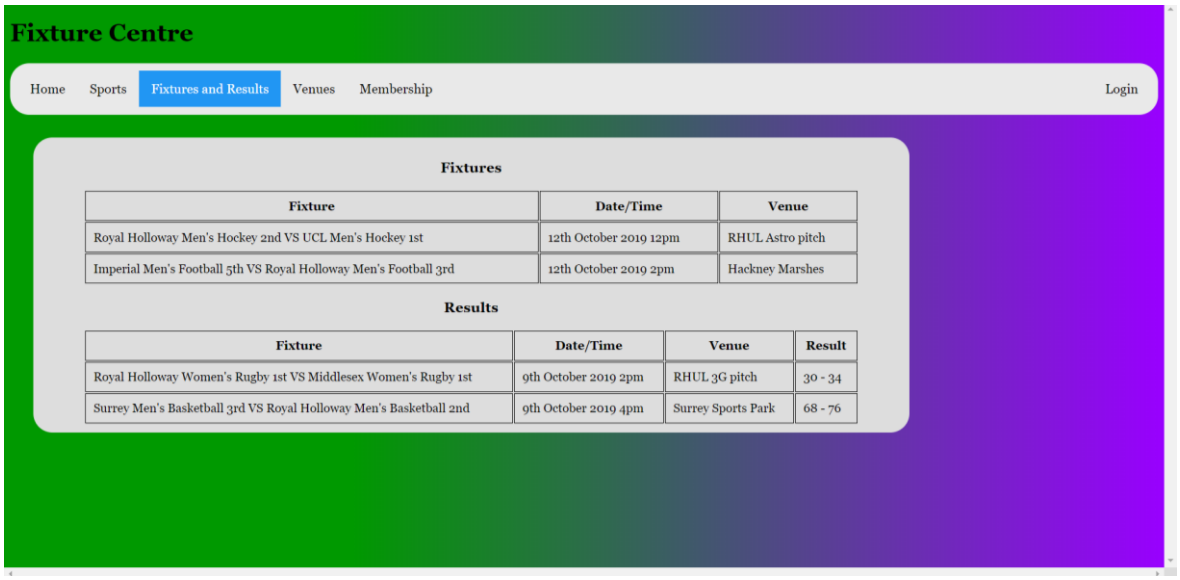


Figure 18. Fixtures and Results page