Interim Project Report

**Full Unit – Interim Report**

A study in (HCI) human computer interaction

Ramnik Dhoot

A report submitted in part fulfilment of the degree of

**BSc (Hons) in Computer Science**

**Supervisor:** Emma Lieu

A logo of a university

Description automatically generated

Department of Computer Science

Royal Holloway, University of London

Table of Contents

[Abstract: 4](#_Toc161348905)

[Project Specifications: 5](#_Toc161348906)

[1. Introduction 6](#_Toc161348907)

[1.1. Aims and Objectives 7](#_Toc161348908)

[1.2 My UI’s: 7](#_Toc161348909)

[1.2.1 Target Audience 8](#_Toc161348910)

[1.3 Motivation: 11](#_Toc161348911)

[1.3.1 What makes this topic so significant: 12](#_Toc161348912)

[2. Literature review: 13](#_Toc161348913)

[In this literature review section, I focus on the primary resources that informed my research, although numerous additional sources, including academic papers and web articles, were also consulted. 13](#_Toc161348914)

[2.1 Human-Computer Interaction (3rd Edition), Alan Dix, Gregory D Abowd, Janet E Finlay and Russell Beale (1): 13](#_Toc161348915)

[2.2 Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition) by Ben Shneiderman (5): 14](#_Toc161348916)

[2.3 Empirical Research in HCI (Human-Computer Interaction: An Empirical Research Perspective by I. Scott MacKenzie) (6): 15](#_Toc161348917)

[3. HCI Theory 16](#_Toc161348918)

[3.1 Background Theory: 16](#_Toc161348919)

[3.2 Models in HCI: 17](#_Toc161348920)

[3.3 Colour Theory 20](#_Toc161348921)

[3.4 Nielsen's usability heuristics 22](#_Toc161348922)

[3.5 Emotional Design 22](#_Toc161348923)

[3.6 Social impact 23](#_Toc161348924)

[3.6 Empirical Research Methods (6) 23](#_Toc161348925)

[HCI Goals for my Ui’s: 24](#_Toc161348926)

[4. Software Engineering 26](#_Toc161348927)

[4.1 Use of Revision Control System 26](#_Toc161348928)

[4.2 Test-Driven Development (TDD) 27](#_Toc161348929)

[4.3 User-Centred Design (UCD) 27](#_Toc161348930)

[4.4 Code Quality Tools 28](#_Toc161348931)

[4.5 My process of design: 28](#_Toc161348932)

[4.6 Design patterns 29](#_Toc161348933)

[4.7 Other software engineering 31](#_Toc161348934)

[4.7.1 Automated testing and CI/CD 31](#_Toc161348935)

[4.7.2 Refactoring 31](#_Toc161348936)

[4.7.3 Documentation 32](#_Toc161348937)

[5. Technical Decisions 32](#_Toc161348938)

[5.1 Technologies I have used and why: 32](#_Toc161348939)

[5.1.1 Java Swing: 32](#_Toc161348940)

[5.1.2 HTML/CSS/JS with Bootstrap: 33](#_Toc161348941)

[5.1.3 React 34](#_Toc161348942)

[6. Final Deliverables 35](#_Toc161348943)

[6.1 My UIs - Designs and HCI principles 35](#_Toc161348944)

[Inventory management system: 36](#_Toc161348945)

[Notes application: 39](#_Toc161348946)

[Shopping website 42](#_Toc161348947)

[6.2 Other HCI methodologies I have included in my designs: 46](#_Toc161348948)

[7. Professional issues 47](#_Toc161348949)

[7.1 Correct Citation and Licensing 47](#_Toc161348950)

[7.2 Accessibility and Usability 47](#_Toc161348951)

[7.3 Professional Issue of Concern in My Project: 47](#_Toc161348952)

[8. Project Analysis 48](#_Toc161348953)

[8.1 Timeline 48](#_Toc161348954)

[8.2 Self Evaluation 52](#_Toc161348955)

[Conclusion 53](#_Toc161348956)

[Appendix 53](#_Toc161348957)

[Diary: 53](#_Toc161348958)

[Bibliography and citations 54](#_Toc161348959)

Abstract:

Human-computer interaction (HCI) is a very diverse field that studies the design and evaluation of interactive computer systems for human use. I'll provide a basic introduction here and delve deeper into details later, ensuring you grasp the report's contents progressively. The goal of HCI is to improve the interaction between users and computer systems by providing the methodologies, principles, and processes for designing interfaces, teaching techniques for evaluating these interfaces, and developing new techniques to improve these interfaces. HCI focuses on three more major concerns; the people, the computers and the interaction between them (the tasks that are performed). Firstly, in HCI you need to understand the user before you can begin designing for them. This involves understanding the user's goals, tasks, and needs, as well as their cognitive, and physical abilities to perform their desired task, the book Human-computer interaction by Alan Dix [1] says that an interaction between a computer and user should be “seamless with respect to their everyday work”.

Another factor of HCI is designing for different types of people and different issues that would arise, for example, what may be simple to understand for a young person who has been accustomed to computers, may not be as easily usable to older people who have had less experience with technology. This would be an issue of accessibility. By understanding the end-user, we can design and implement systems and processes that are easy to use and efficient and we can help facilitate effective communication between users and technology.

Some of the most important principle of HCI are usability and functionality. Usability is a measure of how easy and efficient a system is to use. A usable system is one that can be learned and used quickly and easily, and one that allows users to complete their desired task as expected and effortlessly. Functionality is the services the system provides to the user and the paper [3] explains how the value of functionality is only visible when the system is efficiently utilized by the user. Sometimes it is hard to distinguish what should be more important in a system because if a system had a lot of functionality and isn’t useable then it is useless and vice versa. So, there are levels to usability and functionality and there are trade-offs, for example in my interface I have had to Simplify the navigation of my website which reduces the number of clicks needed to access essential features and looks more visually appealing. While this may enhance usability, it limits the visibility or accessibility of certain advanced functions which could decrease efficiency for higher level users.

Because of this, system designers should prioritize understanding the end user's intentions and their approach to tasks. Overlooking the needs of end users, especially in critical systems such as healthcare or transportation, can lead to design flaws that compromise safety. For instance, poorly designed user interfaces in medical devices or aviation control systems could lead to misinterpretations or errors, directly endangering people’s lives by increasing the likelihood of accidents or critical mistakes.

Alan Dix [1] talks about how aspects such as psychology, computer science, design, and engineering are all intertwined into HCI and how a good designer would have knowledge of the user’s skills and capabilities and would be able to clearly comprehend the users' needs, preferences, and behaviours.

In this project I will explore how crucial a systems design is and I will create 3 user interfaces using various principles and methods I have gathered from all the resources I have gone through then I will demonstrate these principles in my Ui’s. I will also discuss the hardware aspect of a computer and evaluate how usable it is regarding my project. I will also explain the projects aims, professional issues, an evaluation of all the work I have done and the theory I have learnt about HCI.

During the planning phase [2] helped me decide which interfaces I should create, the 3 interfaces I am creating are a shopping website using HTML, CSS, and JavaScript (web-based interface), a notepad desktop application using java and Swing (Graphical user interface), and Database Management System (DBMS) Interface using MySQL for the database and a website for a GUI. They will all be designed using a user-centred design methodology where the user's needs, goals, and preferences are at the forefront of the design process.

Project Specifications:

**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 websites 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 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**

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

**Final Deliverables**

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

1. Introduction

This report outlines the development and findings of my final year project, which focuses on applying Human-Computer Interaction (HCI) principles in designing three distinct user interfaces: a shopping website, an inventory management system, and a note-taking application. The aim is to enhance user’s experiences by making these interfaces intuitive, efficient, and enjoyable to use.

The project is grounded in HCI theory, exploring its key concepts and methodologies to inform the design process. This report includes a comprehensive survey of relevant literature and resources, ensuring a solid theoretical foundation. It delves deeply into the various aspects of Human-Computer Interaction (HCI) that I have explored and learned throughout my research. I will also address the challenges encountered during the project, including technical obstacles and design decisions, and how these were overcome. Then I will discuss my development methodology and software engineering tools which were adopted to complete my proof-of-concept programs in a structured format. All of my programs were created using iterative refinement and prototyping based on testing and evaluation feedback which was crucial for meeting user needs and enhancing usability.

The three user interfaces will each designed for a specific user groups and task, demonstrating the practical application of a range of different HCI methodologies and principles. I will also detail the design process, from planning and development to user testing and final implementation, highlighting the integration of HCI theory and practice.

# 1.1. Aims and Objectives

The aims of this project centre on the design and implementation of three distinct user interfaces (UIs), with a primary focus on the UIs design itself. Then I seek to evaluate these interfaces with a particular emphasis on usability, efficiency, and user satisfaction. Additionally, a key objective is to have a comprehensive understanding of HCI principles and their practical application in the design of interactive systems.

#### These are some specific objectives I have outlined for this project:

thorough research into HCI principles and various UI technologies to establish a solid foundation for the design and implementation phases.

Employ the acquired knowledge to design three distinct UIs, each tailored to meet the specific needs of different user groups and tasks. Also learn the technologies needed such as REACT for database website.

Apply software engineering principles during the creation of my UIs to ensure they are developed efficiently, adhering to industry standards.

Conduct usability testing to evaluate the effectiveness of the UIs. Analyse the results and make necessary adjustments to enhance usability, efficiency, and user satisfaction.

Write a Comprehensive Report: Document the entire design process, including research, design decisions, implementation details, and the evaluation outcomes. The report will also highlight the application of HCI principles throughout the project.

# 1.2 My UI’s:

#### Notes application:

For the note-taking desktop application, I chose to utilize Swing, a java library that will help me make a graphical user interface. Swing was a new technology I had to learn but since I already had experience with java it was much easier to learn that it would have been otherwise, it took me about two weeks of learning, dedicating a small amount of time every day, before I began creating my UI. During this time, I used the official Swing documentation (2) and tutorials, supplemented by an instructional course found on YouTube. I also used material UI Swing to help with the look and feel of the application (19).

This application is specifically designed with the 14-21 age demographic in mind, incorporating Human-Computer Interaction (HCI) principles tailored to this audience. It is a simple note taking application with limited functionality as I am focusing on the design and HCI features but it retains key features commonly found in standard word processing software like the abilities to save, load, delete, and alter the text. Beyond these core functionalities, the application is enhanced with additional usability features such as a customizable sidebar which users can put functions of their own choosing for easy access. The layout and design of the interface have been designed to be intuitive, ensuring ease of learning and use for all users.

#### Shopping Website

For the shopping website, I used HTML/CSS/JS and Bootstrap (17), and the primary focus was on providing an inclusive shopping experience tailored to users with disabilities, particularly those with visual impairments. For this I had to learn how to use JavaScript which did not take too long and I learnt while coding the interface. The website's user experience (UX) design prioritizes simplicity and ease of use simplicity through factors such as straightforward navigation to accommodate use alongside screen readers effectively. This design centres around accessibility, incorporating features such as semantic HTML for meaningful structure, clear and logical navigation paths, different visibility modes like high contrast visuals for enhanced readability, and comprehensive keyboard navigability for those unable to use a mouse.

The development journey began with the establishment of a minimalist interface, primarily navigated through well-organized menus. This foundation allowed for the iterative incorporation of features aimed at enriching the user experience without compromising on accessibility. In committing to these design choices, I meticulously followed the Web Content Accessibility Guidelines (WCAG) 2.1, aiming to achieve Level AA compliance (13). This shows my commitment to making my website a more accessible place for individuals with visual impairments.

#### Database Interface

The Database Interface serves as an inventory management system, made using REACT (15) with HTML, CSS, JavaScript and Bootstrap. This system is specifically designed with business owners in mind, particularly those managing a warehouse and requiring a streamlined method to maintain an inventory of their items. React as also a new technology for me to learn and I had to also learn JavaScript during this time as well for two of my UIs so this learning phase spanned approximately three weeks, I will discuss why I chose to spend this time leaning and using react later.

Some of the key features of this is that this interface would require adaptability across a diverse range of devices, including smartphones, tablets, laptops, and desktop computers. This multi-device functionality caters to the dynamic nature of business operations, ensuring that business owners have constant, flexible access to their inventory data regardless of their location or the device in hand. Also to achieve an interface that is intuitive and functional, I employed a simplistic navigation design.

The incorporation of Bootstrap played a crucial role in this aspect; its responsive design capabilities not only enhanced the aesthetic appeal of the interface but also ensured optimal layout control across various screen sizes and resolutions. This article [7] provided invaluable guidance in designing this interface.

### 1.2.1 Target Audience

In this section I will discuss the specific user groups each of my user interfaces are designed for and analyse their needs and how these groups are expected to interact with the systems. Understanding the target audience is crucial for tailoring the design and functionality of each UI to meet the needs and expectations of its users effectively.

#### Shopping website: Younger/older Visually Impaired Users:

The audience for this website would be users with disabilities, focusing on visual impairments but it would also be used by widely varying age groups so I will discuss the differencing abilities and needs for each of them.

|  |  |  |
| --- | --- | --- |
| Younger | Older | Statistics |
| Younger visually impaired individuals are often more familiar with technology, including screen readers, magnification software, and other assistive technologies. | Older individuals may have varying levels of proficiency with technology, from those who are comfortable using assistive technologies to those who may need simpler interfaces. | Younger users exhibit high adaptability to new technology interfaces​​. Older adults face challenges such as a lack of confidence and physical difficulties in using devices​​.  [Barriers to adoption and attitudes towards tech among older Americans | Pew Research Center](https://www.pewresearch.org/internet/2017/05/17/barriers-to-adoption-and-attitudes-towards-technology/) |
| Often engage in online shopping for a variety of products, including fashion, electronics, and entertainment. | They may visit the website for essential tasks such as online shopping for daily necessities, accessing health information, and staying informed about news and events. | younger generations are heavily engaged in digital platforms for a variety of activities, but older adults are increasingly engaging online.  <https://www.pewresearch.org/short-reads/2019/09/09/us-generations-technology-use/> |
| Would require a modern, simpler and more visually appealing design to resonate with the preferences of younger users. | Provide clear instructions and assistance throughout the website, especially in areas like form filling and checkout processes. | Younger people have a preference for modern and visually appealing designs, based on general trends in technology use.  <https://www.pewresearch.org/short-reads/2019/09/09/us-generations-technology-use/> |
| The website should be responsive and compatible with various devices, including smartphones and tablets. Older users also use their smartphones for shopping online. | | 96% of those aged 18 to 29 own a smartphone​​, 61% of those 65 and older own a smartphone.  <https://www.pewresearch.org/short-reads/2022/01/13/share-of-those-65-and-older-who-are-tech-users-has-grown-in-the-past-decade/> |

Visually impaired users navigate and interact with digital content in ways that differ significantly from users without visual impairments. Their experience and interaction with technology are done by relying on alternative senses and tools to access information. For instance, where a sighted user might quickly scan a webpage for information or click on a visually distinct button, a visually impaired user would depend on screen readers to read out the text content of the page, including navigation menus and alt text descriptions of images. This reliance on auditory feedback to navigate means that website layouts, menu structures, and even the choice of words used in links and buttons must be clear and logically organized.

In designing interfaces for visually impaired users, it's crucial to recognize the spectrum of visual impairments, from mild to complete blindness. My designs are crafted to accommodate this diversity, employing adaptable features and compatibility with assistive technologies to ensure a seamless and inclusive user experience.

Visually impaired users may use voice recognition software to input commands or type text, this can include navigating to different sections of a site, filling out forms, or performing searches, which requires my website to support voice input effectively. The use of refreshable braille displays is also a new technology that allows users to 'read' the screen through tactile feedback, it converts on-screen information into braille characters that can be felt with the fingers, providing a direct and tactile way to interact with text on a digital device. My UI is also designed with this and other technologies in mind. Understanding these unique interaction methods and challenges is crucial for developers and designers to create more accessible and inclusive digital environments.

#### Inventory management system: Entrepreneurs and Small Business Owners/ managers.

Business managers and entrepreneurs who use inventory management systems span a diverse range of business types, including retail stores, e-commerce ventures, manufacturing companies, and service providers. This diversity necessitates a system design that is intuitive and accessible to users with varying levels of technological expertise. Ensuring that the system is user-friendly and straightforward allows business owners to navigate and utilize the system effectively, regardless of their prior experience with similar technologies, this not only serves as a valuable asset for their operations but also has the potential to enhance their profitability.

Time efficiency is another critical consideration for business owners, who often are multitasking by managing various aspects of their operations simultaneously. They require an inventory management system that enables them to perform tasks and access vital information quickly as needed. The ability to quickly perform actions such as check stock, and place orders can significantly impact the daily operations of a business. As such, my interface must be designed for speed and efficiency, enabling entrepreneurs to manage their time more effectively.

Some other considerations for business owners are the flexibility to adjust the system as needed allows businesses to optimize their operations and align the inventory management process with their strategic goals. Security and privacy are also very important given the sensitive nature of business data. Entrepreneurs prioritize systems that can safeguard their information against unauthorized access and breaches. This includes secure login mechanisms, data encryption, and regular security updates to protect against emerging threats. Trust in the system's security measures encourages business owners to confidently rely on it for their inventory management needs. My UI is designed to facilitate the needs of business owners and to help them strive in their business endeavours.

#### Notes application: Students

Students, ranging from those in high school to those in college, have specific needs when it comes to using a notes application, primarily due to their academic focus. Younger students, for example in secondary school, are engaged in multiple different subjects at the same time. This diversity in subject matter necessitates features within the notes application that support varied study methods and content types. For instance, tools that facilitate revision, such as flashcards and summarization features, along with reminders for important dates and deadlines, would be highly valued. These functionalities help students efficiently organize their study material and manage their academic schedules. As students’ progress and specialize in specific fields of study, their needs become more sophisticated, requiring the notes application to offer advanced features tailored to their coursework and research activities. For example, the ability to integrate multimedia elements, specialized formats for different types of data, and collaboration features for group projects.

Given that students are generally tech-savvy but may have varying attention spans, the user interface (UI) of the note’s application needs to be visually engaging and intuitive. A clean, easy-to-navigate UI with clear and concise information presentation can significantly enhance usability, particularly for students who spend long hours studying and require quick access to their notes. These considerations are crucial in developing my application to ensure it is highly usable and supportive of users' academic pursuits.

# 1.3 Motivation:

#### Why I’m interested:

This project presents a unique opportunity for me to engage in the creation of three distinct UIs tailored for diverse user groups and tasks. The prospect of delving into various UI technologies is very good for me as it will broaden my skill set and empower me to make UIs that seamlessly blend usability with aesthetic appeal.

Furthermore, the project's emphasis on usability testing is very important as the ability to conduct tests and derive meaningful insights from user feedback is a valuable skill I will need in any of my future endeavours. This hands-on experience will not only enhance the practicality of my UIs but also deepen my understanding of the user experience.

This HCI project offers a comprehensive exploration of a wide array of HCI principles, ranging from cognitive psychology to design principles and evaluation methods. Acquiring knowledge in these diverse areas and subsequently applying them to create interactive systems that are both user-friendly and effective is a compelling aspect of this project. Overall, I am enthusiastic about the learning opportunities this HCI project affords, allowing me to blend creativity with a solid understanding of human-computer interaction.

#### How this will help me in my future career:

Engaging in this HCI project will significantly contribute to my future career development in multiple ways. Firstly, it will provide me with a thorough understanding of HCI principles and their practical application in designing interactive systems. This knowledge is invaluable, as it forms the foundation for creating user interfaces that are not only functional but also user-friendly and effective.

Secondly, the project will equip me with practical skills in UI design, implementation, and evaluation which are increasingly sought after in the job market. As companies prioritize user-centric approaches, the ability to create the best interfaces for users becomes a crucial asset. Moreover, completing this project will help me get a range of skills that are vital in any professional setting. The problem-solving skills developed through HCI, which is a very problem-solving based discipline, will enhance my ability to identify and address challenges effectively. My critical thinking skills will be improved as I engage with evidence-based practices in HCI, enabling me to evaluate research findings and make better decisions in any task I have.

The creative aspects of HCI will nurture my ability to generate innovative solutions to design problems and lastly, the project will further enhance my research skills, providing a solid foundation for evidence-based decision-making in my future career. So, this HCI project serves as a comprehensive learning experience that not only equips me with the technical skills demanded in the field but also teaches me skills crucial for success in my future jobs.

### 1.3.1 What makes this topic so significant:

The significance of HCI for UI designers lies in its direct impact on our daily lives. As we engage with computer devices on a regular basis, the quality of these interactions is crucial. HCI principles serve as a guide for UI designers, aiming to make these interactions as efficient, effective, and enjoyable as possible. HCI principles play a pivotal role in creating user interfaces that align with user expectations, emphasizing ease of learning and use. Digital interactions have become integral to our routines; users seek interfaces that allow them to complete tasks swiftly and effortlessly and designers can craft interfaces that not only meet these expectations but also enhance the overall user experience.

Furthermore, HCI principles address the desire for intuitive interfaces. Users prefer interactions that do not necessitate reading manuals or mastering complex actions. UI designers, by incorporating HCI principles, can create interfaces that are not only user-friendly but also intuitive and easy to comprehend. This is particularly significant as it aligns with the user-centric approach, ensuring that digital tools are accessible and usable for a diverse range of users. So HCI provides the ability to bridge the gap between technology and users' expectations and by using HCI principles, designers can create interfaces that not only meet functional requirements but also contribute to a positive and seamless user experience, ultimately making technology more accessible and enhancing usability.

#### Uses and applications of HCI:

Human-Computer Interaction (HCI) holds significant importance in various domains, in the realm of medical devices, HCI plays a crucial role in ensuring the safety and user-friendliness of these devices, catering to the needs of both patients and healthcare professionals. In educational software, HCI is instrumental in creating engaging and effective interfaces for students, optimizing the learning experience. For websites and mobile apps, HCI principles contribute to seamless navigation and user-friendly interfaces, ensuring that users can effortlessly interact with digital platforms. (20) By applying HCI principles across these diverse applications and systems, designers can tailor interfaces to meet specific user needs, ultimately enhancing the overall usability and satisfaction of users in various contexts. (21)

2. Literature review:

### In this literature review section, I focus on the primary resources that informed my research, although numerous additional sources, including academic papers and web articles, were also consulted.

### 2.1 Human-Computer Interaction (3rd Edition), Alan Dix, Gregory D Abowd, Janet E Finlay and Russell Beale (1):

This book introduces the core concepts and scope of HCI, it defines HCI and its importance in the modern world. It is a foundational text that offers comprehensive insights into HCI principles, design considerations, and user experience. The book's focus on user-centred design, iterative development, and robust evaluation techniques makes it invaluable in the HCI field. I will be discussing the most important chapters that pertain to my UI’s here and their significance in my UI’s.

In the first three chapters, the authors provide an overview of the human and computer aspects of HCI, as well as the interaction between them. This is all explained through cognitive Models (also further discussed in chapter 12) and models of interaction. Alan Dix discusses the use of cognitive models to understand how users process information and interact with computer systems. He also discusses other models like the Human and The Computer and how you need to understand these to be able to create a good and usable UI. These models can inform the design of interfaces that are more aligned with human cognitive processes, making them easier to use.

The book strongly advocates for UCD as a framework for HCI development. UCD involves users throughout the design process through techniques such as user research, persona creation, and usability testing. This approach ensures that the final product is tailored to the users' needs, preferences, and limitations.

Another key methodology discussed is the iterative design process. This involves repeatedly designing, prototyping, and testing interfaces, then using the feedback to refine the design. This cycle helps in identifying usability issues early and ensures that the final interface is as intuitive and user-friendly as possible.

Chapter 7 is one of the main chapters that helped me design my UI’s as it talks about design rules and designing for maximum usability. This is where the book starts to delve into abstract principles of design to help the reader understand how to design for usability. It also discusses how these design rules and principles help form standards and guidelines to provide direction when designing a UI. Chapter 16 also helped with this as it talks about the design of dialog and Notations used for dialog description for example, Input forms can be can being diagrammatic or textual. This chapter helped me learn how instructions can be effectively portrayed to the user as well as which interaction styles should be used such as menus and forms.

Some other topics this book delves into are evaluation techniques to test the usability and functionality of a system and approaches towards this (Chapter 9). Chapter 10: Universal design is about “designing systems so that they can be used by anyone in any circumstance” and also discusses multi-modal systems which have various methods of inputs, which was important for my notes application which is also a touchscreen app. This chapter was especially important to me as here it discusses how we should design for diversity such as people with sensory, physical or cognitive impairment, people of different ages and people from different cultures and backgrounds.

Overall, Human-Computer Interaction (3rd Edition) is an excellent resource for anyone interested in HCI. The book is well-written and provides a good balance between theory and practice. The authors provide many examples and case studies to illustrate the concepts covered in the book, making it easy to understand and apply the material. Its comprehensive overview and its emphasis on core HCI principles make it a valuable resource for students, practitioners, and researchers and Its focus on user-centred principles ensures its continued relevance as a key reference in the field.

### 2.2 Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition) by Ben Shneiderman (5):

This book is an essential text in the field of human-computer interaction, providing an in-depth exploration of design principles and strategies crucial for creating effective user interfaces. This comprehensive overview encapsulates the methodologies for designing interfaces that are efficient, learnable, and satisfying for users, it also gives practical examples and case studies that illustrate these concepts in action.

Chapter 1, titled "The Eight Golden Rules of Interface Design," explains Shneiderman's foundational guidelines for crafting user-centric interfaces. These rules, which include striving for consistency, enabling shortcuts, offering informative feedback, and minimizing users' memory load, are instrumental in enhancing the user experience by making interfaces more predictable, forgiving, and easy to navigate.

the book delves Direct Manipulation in chapters 2 and 5, which is a design strategy where users interact with on-screen objects in a way that mimics the manipulation of physical objects. This approach aims to make systems more intuitive and engaging by providing immediate feedback to actions, thereby enhancing the user's sense of control and ease of use. Shneiderman's advocates for a UCD that prioritizes the needs, wants, and limitations of end-users throughout the design process. He also outlines methods such as user research, persona creation, and usability testing, ensuring that the final product resonates with the target audience's preferences and requirements.

Information Visualization focuses on strategies for presenting complex data through visual representations like graphs and charts. This is pivotal for enabling users to comprehend and interact with information, thereby aiding in better decision-making and insight generation. This was very helpful for me when creating the database interface as I had a lot of information which I had to display visually. This book also discusses Universal Usability and the significance of creating interfaces accessible to a diverse range of users, including those with disabilities. Strategies for achieving universal usability, such as adherence to accessibility standards and inclusive user testing, are discussed to ensure designs cater to a broad spectrum of user capabilities and preferences. These were invaluable in designing my shopping website.

In conclusion, "Designing the User Interface" by Ben Shneiderman offers a profound framework for developing human-computer interfaces that are not only functional and efficient but also enjoyable and accessible to a wide audience. Through its emphasis on user-centred design, direct manipulation, and the eight golden rules, the book provides a robust foundation for meeting and exceeding user expectations.

### 2.3 Empirical Research in HCI (Human-Computer Interaction: An Empirical Research Perspective by I. Scott MacKenzie) (6):

Scott MacKenzie's 'Human-Computer Interaction: An Empirical Research Perspective' offers a comprehensive exploration of empirical research methodologies. The book sets itself apart by focusing on the scientific principles essential for evaluating and designing computer interfaces and interaction techniques, distinguishing itself from more design oriented HCI literature like the previous two books.

The initial chapters provide a foundational overview of HCI's historical context, the human factor in technology use, interaction elements, and the basics of scientific inquiry and research methodology. This grounding prepares readers for the subsequent, more detailed exploration of empirical research methods in HCI. Chapter 1, Introduction to Empirical Research, detailing the historical development of HCI and introducing the key concepts. It sets the stage for the empirical research perspective that the book advocates.

Then it dives into the core scientific methods applicable in HCI research. This includes hypothesis formation, the design and execution of experiments, and the ethical considerations crucial for research involving human participants. MacKenzie emphasizes the importance of research ethics, informed consent, data privacy, and minimizing harm. In Chapter 3 explores various experimental designs (between-subjects, within-subjects, and mixed designs), detailing how to control variables and conduct statistical analysis to ensure robust and valid research findings.

It also discusses quantifying interaction effectiveness through metrics such as speed, accuracy, error rates, and subjective user ratings. This chapter introduces Fitts' Law as a predictive model for HCI and discusses qualitative assessments for gathering in-depth data on user experiences. Chapter 5 guides readers through designing and implementing HCI experiments. It covers hypothesis formulation, apparatus and material selection, participant recruitment, and procedure design, including counterbalancing techniques to address order effects combined with a thorough overview of statistical analysis methods, data visualization techniques, and the intricacies of writing and publishing research papers in the HCI field.

MacKenzie's text is invaluable for its rigorous approach to empirical research methods in HCI, ensuring that studies in the field are founded on reproducible and sound findings. However, the book's emphasis on empirical research might limit its direct applicability for me as a designer as I am not primarily engaged in research. The inclusion of advanced research techniques highlights the depth of investigation required to enhance human-computer interactions, making this text a cornerstone for those committed to advancing the field of HCI.

3. HCI Theory

# 3.1 Background Theory:

Human-Computer Interaction (HCI) is a diverse field that explores the design, development, and use of computer systems from the perspective of the user. This field draws upon principles from computer science, cognitive psychology, design, ergonomics, and sociology to enhance the usability, accessibility, and experience of computer systems (1). It may also be said that it is a study of how humans use computers to perform certain tasks and use it in such a way that the interaction is being enjoyed and effective.

Some of the most important aspects of HCI are, Cognitive psychology which focuses on how humans think, learn, and remember information. It is important for understanding how users interact with computers and how to design UIs that are easy to learn and use (4). Design principles are general principles that can be used to design effective user interfaces. Evaluation methods are methods that can be used to evaluate the usability of UIs. Some of the most common evaluation methods include user testing and reviews (6).

HCI is important because it helps to ensure that interactive systems are designed to be usable, efficient, and enjoyable to use (20). This is important for several reasons, including Increased productivity, reduced errors, usable systems can help to reduce the number of errors, and increased satisfaction. So, the goal of HCI is to create user interfaces and experiences that are not only efficient and functional but also intuitive, enjoyable, and accessible.

At the core of HCI theory is the principle of User-Centred Design discussed in (1). By involving users in the design and evaluation phases, developers can create interfaces that align with users' mental models and expectations. These are some examples of essential HCI principles:

**Usability** is a fundamental concept in HCI, focusing on the effectiveness, efficiency, and satisfaction with which users can accomplish tasks within a system. Usability principles guide the design of interfaces to minimize errors, streamline workflows, and enhance overall user experience.

**Affordances** refer to properties of an object that suggest how it can be used. Signifiers are cues or indicators that communicate these affordances. In HCI, understanding and appropriately implementing affordances and signifiers help users intuitively grasp how to interact with a system.

**Feedback and Guidance:** Providing timely feedback to users about the outcome of their action, contributes to a sense of control and predictability, enhancing the user's experience.

# 3.2 Models in HCI:

Donald Normans model of interaction (14) outlines the basic stages of interaction between a person and a computer: forming the goal, forming the intention, specifying the action, executing the action, perceiving the system state, interpreting the state, and comparing the outcome with the goal. This model has informed countless HCI designs by highlighting the importance of understanding user intentions and how they translate into actions within a system.

Alan Dix (1) also talks models in HCI and about “the golden rule of design”: which is understanding your materials. Understanding computers – limitations, capacities, tools, platforms and Understanding people – psychological, social aspects, human error. I will discuss what these models mean for designers.

#### 1. The Human:

This model aims to represent human cognition and how users process information when interacting with computer systems. These models help designers predict user behaviour, identify potential usability issues, and devise interfaces that align with human cognitive capabilities.

For understanding

humans as an information-processing system, how they communicate, characteristics of the human/user as a processor of

information- Memory, attention, problem-solving, learning, motivation, motor skills, conceptual models and diversity.

Language, interaction and communication -

Aspects of language-Syntax, pragmatics, semantics, conversational interaction and specialized languages.

This model breaks down human processing into three subsystems: perceptual, cognitive, and motor, each with its own cycle time and this aids in predicting how design changes might impact user performance. Here, the human mind is conceptualised as an information processing system, providing insights into how humans perceive, process, and respond to information.

How do users think? Users appreciate quality and credibility. For example, Users don’t generally read, they scan:



These images depict heat maps revealing the focal points of user attention during online browsing. The most viewed areas, represented as the "hottest" zones, tend to cluster around the middle of sentences, which aligns with the typical scanning behaviour of web users. A user’s impatience and the pursuit of instant gratification drives them to satisfice, opting for the first reasonable choice instead of the optimal one. Optimization is a lot harder and takes a lot more time, so satisficing is generally the more efficient approach. Users rely on their intuition and seek control in their interactions.

A fundamental principle in creating a user-friendly interface is to minimize cognitive effort, following Krug's first law of usability (22). Webpages should be clear, self-explanatory, and have clear structure, with moderate visual cues and easily recognizable links. By reducing cognitive load, you make it easier for visitors to grasp the idea behind the system. Recognizing users' limited patience, designers should strive to keep user requirements minimal, the less action is required from users to test a service, the more likely a random visitor is to try it out.

Effective management of users' attention is crucial. Web-users can instantly recognize edges, patterns and motions, a webpage should guide attention to specific areas through use of visual elements. This approach can help your visitors to get from point A to point B without thinking of how it is supposed to be done. Guidelines are extremely effective as they lead the visitors through the site content in a very simple and user-friendly way. The less confusion and questions a user have about the page, the better the sense of orientation and provides an overall improved user experience. (11)

Mental models in refer to the internal representations that users form based on their beliefs and understanding of how a system works. These models are fundamental as they significantly influence how users predict the behaviour of a system and, consequently, how they interact with it. The efficacy of a user's mental model directly impacts their ability to navigate and utilize a system efficiently, showing the importance of these cognitive constructs in the design and development of user interfaces.

One of the biggest challenges within HCI is the alignment of the designer's mental model with that of the users. This alignment is critical because a mismatch between the two can lead to confusion, errors, and a decrease in overall system usability. Designers often possess an in-depth understanding of the system's mechanisms and logic, which can inadvertently lead to assumptions that users share this level of comprehension. However, users typically do not have the same background knowledge or technical expertise, which can result in a disconnect between how designers envision the system being used and how it is actually used in practice.

To bridge this gap, it's essential for designers to understand their end-users, incorporating feedback early and often throughout the design process. Techniques such as user testing, prototyping, and iterative design play crucial roles in revealing target user’s mental models and adjusting the system accordingly. By striving for a close alignment between the user's and the designer's mental models, HCI professionals can create more intuitive, accessible, and user-friendly interfaces that meet users' needs and expectations, leading to a more effective and satisfying interaction with technology.

#### The computer:

The Model of the Computer within Dix's framework (1) refers to the representation of the computer's behaviour and functionalities from a user's perspective. It's not about the technical or hardware specifics of the computer system but rather how the system is perceived to operate by the user. This model is crucial for understanding how users predict the computer's responses to their inputs and how these predictions affect their interactions with technology.

Computers are good at counting and measuring, precise storage and recall, rapid and consistent

responses, data processing or calculation, formulations, repetitive actions, and performance over time, “Simple and

sharply defined things”.

Dix emphasizes the importance of ensuring that the computer model is transparent and predictable to the users. This transparency allows users to form accurate mental models of how the computer works, which in turn, facilitates more effective and efficient interactions. For instance, when a user understands that clicking a "save" icon will store their document, this reflects a clear alignment between the user's mental model and the model of the computer as designed by the interface.

To achieve such alignment, Dix advocates for design principles that support consistency, feedback, and affordances, enabling users to learn and adapt to the system quickly. By carefully considering the model of the computer in the design process, HCI professionals can create interfaces that not only meet the users' needs but also enhance their overall interaction experience with the computer system. Furthermore, a system's learnability and adaptability over time influence the users evolving conceptual model of computer systems.

A diagram of a computer science

Description automatically generated

#### Interaction Models in Human-Computer Interaction

Interaction models in HCI provide frameworks for understanding and designing the ways in which users interact with computers. These models are crucial for creating user interfaces that are intuitive, efficient, and effective. Input and output modalities refer to the ways users can provide input to a computer system and receive output from it. Traditional modalities include keyboards and mice for input, and screens for output. Emerging modalities encompass touch, gesture, voice, and even brain-computer interfaces (BCI) for input, as well as augmented reality (AR) and virtual reality (VR) for output.

Direct manipulation, introduced by Shneiderman in the 1980s, is a model where users interact with visible objects on the screen in a manner that feels like physically manipulating the objects. This model emphasizes immediate, reversible actions with continuous feedback, allowing users to feel in control of the computer application.

Donald Norman also outlines the gulfs of execution and evaluation, which represent the gap between a user's goals and the means to achieve them. The gulf of execution refers to how well the system allows users to enact their intentions, while the gulf of evaluation reflects the system's ability to provide feedback that matches the user's expectations. This helps designers consider each step a user must take to accomplish their goals and how each step can be supported by the interface.

#### Interaction Techniques

These techniques aim to make interactions more intuitive, efficient, and aligned with human behaviours and expectations.

Interaction Styles: The book examines a range of interaction paradigms:

* Command-line interfaces: Historical but still relevant in specialized contexts.
* Menus: Structured choices for efficient selection.
* Form fill-in: Focused data input.
* Direct manipulation: Simulating real-world interaction with objects.
* Natural Language: Understanding conversational inputs.

Gestural Interfaces interact with digital systems through body movements and gestures. This includes touch gestures on screens (e.g., pinching, swiping) and air gestures recognized by sensors or cameras. This offers a more natural and intuitive way for users to control devices, particularly in environments where traditional inputs are impractical.

Voice user interfaces enable interaction with systems through spoken commands, making technology accessible without the need for physical inputs. VUIs are central to digital assistants like Siri, Alexa, and Google Assistant and they can significantly improve accessibility for users with physical disabilities and provide convenience in contexts where hands-free operation is preferred.

AR overlays digital information onto the real world, while VR creates fully immersive virtual environments. Both technologies offer rich, interactive experiences that can go beyond the constraints of traditional screens, allowing you to do more tasks than you would otherwise be able to do. AR and VR have applications in education, training, entertainment, and healthcare, offering immersive ways to learn, collaborate, and explore. VR provides an immersive experience and engages users in a fully digital environment, often using head-mounted displays (HMDs). AR provides enhanced reality, it superimposes digital information onto the real world, visible through devices like smartphones or AR glasses.

Diagram of a diagram of a speaker

Description automatically generated with medium confidence

### 3.3 Colour Theory

Colour theory plays a pivotal role in enhancing the interaction between humans and computers, leveraging the psychological and emotional impacts of colours to influence user behaviour and perceptions. Alan Dix talks about how colours can convey meaning, draw attention, group elements, and evoke emotional responses, making their thoughtful application a powerful tool in enhancing usability and user satisfaction.

Dix highlights the use of colour coding to organize information, indicate status, or differentiate between types of data. Colours also carry cultural and emotional meanings that can vary significantly across different groups. Dix advises designers to consider the target audience's cultural background and the emotional impact of colour choices to ensure the interface communicates effectively and appropriately. Techniques such as using texture or patterns in addition to colour, ensuring sufficient colour contrast, and providing alternatives to colour-based information are recommended to make interfaces accessible to a broader audience, especially users with colour vision deficiencies.

Complementary colours, positioned opposite each other on the colour wheel, create the strongest contrast and this can be used to highlight important elements on a user interface, such as call-to-action buttons or critical notifications, making them stand out against the rest of the design. Analogous colours, which are adjacent to each other on the colour wheel, foster a sense of harmony and unity creating a calm and cohesive atmosphere within the interface, ideal for applications aiming to offer a relaxing user experience.



Incorporating Triadic colour schemes involve colours that are evenly spaced around the colour wheel, contributing to a sense of balance and stability. This approach can give a professional and sophisticated appearance to the interface, suitable for business or productivity applications. Utilizing Tetradic colour schemes combine two complementary pairs, blending both contrast and harmony. This dynamic use of colours can generate interest and excitement, capturing users' attention effectively. It's particularly useful in designs aiming to be engaging and vibrant. Although employing a limited colour palette tends to be more impactful and less overwhelming for users, facilitating a clearer and more focused user experience.

By applying these principles of colour theory in HCI design, developers and designers can create more engaging, accessible, and effective user interfaces that not only appeal visually but also enhance usability and user satisfaction. However, a reliance on colour can be detrimental to users who are colourblind or have difficulty distinguishing certain colours. For example, the most prevalent form of colour-blindness is red-green colour-blindness so designers should ensure their interfaces don’t rely solely on red and green hues.

### 3.4 Nielsen's usability heuristics

Nielsen's usability heuristics are a set of principles that can be used to design more usable UI’s. They are based on the idea that users should be able to easily learn, use, and remember how to use a system. The heuristics are designed to help designers avoid common usability problems. (12)

1. Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.
2. Match between system and the real world: The system’s language, terminology, and concepts should align with what users already know from their real-world experiences.
   1. This also links with the HCI principal affordance which refers to the perceived and actual properties of an object that determine how it can be used. For example, A door handle affords pulling, while a button affords pushing. Understanding affordances helps designers create interfaces where users can intuitively guess how to interact with them.
3. User control and freedom: Users should be able to backtrack or undo actions easily, offering them the freedom to correct mistakes.
4. Consistency and standards: Follow platform conventions and be consistent in your use of terminology and design.
5. Error prevention: Careful design that prevents a problem from occurring. Double-check user input and provide clear instructions to avoid mistakes.
6. Recognition rather than recall: Minimize the amount of information that users must remember by providing appropriate defaults or making information easily retrievable.
7. Efficiency and flexibility: Efficiency features and shortcuts should be available for power users without overwhelming beginners. Accommodate both users by providing clear navigation options and the ability to customize frequent actions.
8. Aesthetic and minimalist design: Clutter-free, aesthetically pleasing designs are more user-friendly.
9. Help users recognize, diagnose, and recover from errors: Error messages should be clear, concise, and constructive. They should tell users what the problem is, and how to fix it.
10. Help and documentation: Provide easy access to help and documentation, and make sure it is clear, concise, and up to date.

### 3.5 Emotional Design

Emotional Design is an evolving area in HCI that focuses on the emotional relationship between the users and the systems they interact with. Effective design appeals not only to users' cognitive skills but also to their emotions, creating a more profound and enduring user experience. It operates on the understanding that users are not just rational beings but emotional ones as well, and their interactions with technology can significantly impact their mood, satisfaction, and loyalty to a product. By integrating aesthetic appeal, usability, and personal connection within the design, emotional design aims to elicit specific emotions and foster a deeper bond between the user and the product.

Norman (14) Talks about The Three Levels of Emotional Design:

Visceral Design: This level deals with the initial impact of a product, its appearance, touch, and feel. Norman emphasizes the importance of visceral reactions, stating that “they are about immediate response – do I like this? Do I want it?” This aspect of design is about the sensory experience of using a product.

Behavioral Design: At this level, the focus is on usability and the experience of using the product. Good behavioural design is all about feeling in control, which includes understanding how to use the product and getting feedback. “It’s where function meets form”.

Reflective Design: This is the most complicated level and concerns the user's conscious thought about the product, including its impact on their lives and their self-image. Reflective design is about the meaning of things, the personal and cultural significance of a product.

While emotional design holds great promise, it also presents challenges, particularly in terms of privacy and manipulation. Designers must navigate these ethical considerations carefully, ensuring that their work respects user autonomy and privacy.

### 3.6 Social impact

HCI has transformed the dynamics of social communication and interaction. The development of social media platforms, messaging applications, and video conferencing tools, have bridged geographical gaps, enabling people to have relationships across vast distances and fostering global connectivity (3). However, this digital ease of connection introduces challenges, including the spread of misinformation and the potential negative impacts on mental health resulting from digital communication overuse or dependency (20). These concerns underscore the importance of designing with a conscientious approach to HCI, aiming to mitigate such risks while enhancing social bonds.

HCI's contribution extends to enhancing technology's accessibility for individuals with disabilities by integrating accessible design features, such as voice recognition, screen readers, and alternative input methods. HCI ensures that digital content and services are usable by a diverse range of users. This inclusivity not only empowers a wider audience to partake in educational, professional, and social endeavours but also plays a role in dismantling barriers and cultivating a more inclusive digital environment (21). Furthermore, the designs produced within the field of HCI can actively shape societal values. The choices made during the design process, including the inclusion or exclusion of certain features and the representation of individuals and communities, can greatly affect how users think and act.

### 3.6 Empirical Research Methods (6)

Surveys and Interviews are pivotal in gathering both qualitative and quantitative data regarding user preferences, experiences, and needs. This approach enables a comprehensive understanding of the target audience, which is essential for tailoring design decisions to match user expectations and requirements.

Usability Testing is another crucial component, which entails observing users as they interact with a system. This process is instrumental in identifying any usability issues that could hinder the user experience, allowing designers to make necessary adjustments to enhance the system's overall usability.

Prototyping allows the creation of mock-ups or working models of interfaces. These prototypes serve as exploratory tools to test design concepts and gather feedback from users, facilitating iterative improvement. Additionally, User-Centred Design (UCD) emphasizes the importance of involving users throughout the design process. This methodology ensures that the final product is closely aligned with user needs and preferences, thereby increasing its likelihood of success.

In the realm of Evaluation Techniques, Heuristic Evaluation and A/B Testing are prominent methods. Heuristic Evaluation involves experts using established heuristics to assess the usability of an interface, while A/B Testing involves comparing two versions of a web page or app to determine which one performs better based on specific metrics. These techniques are invaluable for refining and validating design choices.

Evaluation techniques are vital in HCI because they bridge the gap between theoretical design principles and real-world user experiences. They provide the evidence base needed to make informed design decisions, promote inclusivity, and ultimately, drive the creation of digital products that enrich and facilitate human-computer interaction. Without these techniques, the development of user-centric technology would lack direction.

Prototyping: The book covers various prototyping techniques, from low-fidelity sketches to high-fidelity interactive prototypes. Prototyping is presented as an essential tool for exploring design ideas, communicating concepts to stakeholders, and gathering user feedback.

The experiments in hci commonly prefers the younger group as being young, they are technically

knowledgeable, highly educated, and are unrepresentative of demographic realities. In the case of the older people in

research groups, the collection of the data from these participants require alterations and research methods. Formal

education and literacy level are characteristics in which older group of people differ widely from younger group of

participants in the research.

# HCI Goals for my Ui’s:

The HCI principles I have chosen to implement across my user interfaces are detailed here, highlighting the distinctions among the three UIs. These differences are purposeful, tailored to each interface's specific context and user needs, underlining the versatility and adaptability of HCI principles in diverse design scenarios.

#### Shopping website:

Clarity/Usability: The primary goal is to ensure that the shopping website is user-friendly and easy to navigate. Users should quickly find the desired products with minimal effort.

Consistency: Maintain consistency in design patterns and terminology. This will make it easier for users to learn and use the website.

* Mental models of users must be respected, aligning the design with user expectations.

Recognition rather than recall should be emphasized, utilizing familiar design patterns and terminology for a seamless user experience.

Feedback and Response Time: Provide clear feedback for user actions, whether successful submissions or errors. Then guide the user on what to do next.

* Response times should be reasonable so that users don’t get frustrated during interactions.

Error Prevention and Forgiveness: Implement features that help prevent user errors and allowing users to recover easily from mistakes or explore different options without significant consequences.

Visibility and Affordance: Ensure interactive elements are clearly visible with visual cues indicating their functionality.

* Visual hierarchy is essential, using large fonts or contrasting colours to highlight important information.

Accessibility: Integrate accessibility features like keyboard navigation, screen reader compatibility, and alternatives for multimedia content to make it easier for people with disabilities to navigate and use.

* Ensure the website is usable by individuals with visual impairments.

Learnability: Design the website for quick user understanding, minimizing the learning curve for all users, especially for those with visual impairments.

I have also created this with respect to WCAG 2.1. These are guidelines that aim to make online content more accessible to individuals with disabilities. Focusing on principles of perceivability, operability, understandability, and robustness, WCAG 2.1 outlines specific success criteria to ensure websites are usable by people with diverse abilities.

#### Notepad application:

Focus: The notepad application is designed to create a distraction-free environment for writing. Users can focus on their writing without interruptions from any other functionalities e.g., notifications.

Usability: The application should prioritize ease of use, allowing users to quickly locate and utilize its features. Also be intuitive, ensuring a seamless and efficient experience.

Flexibility: Recognizing diverse user needs, the notepad application should have the ability to customize the application to suit user preferences, enhancing the overall experience. E.g. different methods of interacting with the system

Power/Efficiency: The notepad application should be equipped with a range of features to boost user productivity, especially for expert users who require advanced functionalities.

Enjoyment: It should be enjoyable to use seeing as it is targeted towards students. User satisfaction is a key to keep students engaged as they are more likely to have lower attention spans.

Internationalization: For a global user base, the notepad application should adopt a neutral language and design. Cultural references are avoided to ensure universal understanding among users from different backgrounds.

#### Database management system interface:

Learnability/Memorability: Ensure that the database management system interface is easily learnable. Users should quickly grasp how to navigate and utilize the interface efficiently as they are business owners and need to be able to operate the system easily.

* Design the interface to be memorable, allowing users to recall its usage even after a period of non-use.

Efficiency and Productivity: Prioritize the completion of tasks with efficiency and ease. Streamline workflows to empower business owners in managing their inventory effectively, minimizing the steps/complexity needed for commonly done tasks.

Satisfaction: Aim for a positive user experience, instilling confidence in users for using the system

Customization: Allow users to customize and personalize the interface according to their preferences. This is so users can easily deal with their most important tasks.

Feedback/Status Updates: Implement a feedback mechanism to keep users informed of the status of their actions.

Security and Privacy: Facilitate the security of sensitive inventory data by implementing robust authentication measures to safeguard against unauthorized access and protect the confidentiality of information.

Scalability/Robustness: Ensure that the system can handle inventories of varying scales, accommodating businesses with different sizes and levels of complexity. (This is not very applicable because my interface will not have backend functionality, but I will make it with this in mind)

4. Software Engineering

Throughout the development of this project, I have been applying software engineering principles and methodologies to ensure that all my interfaces are not just user-friendly but also robust, reliable, and easy to understand and maintain.

### 4.1 Use of Revision Control System

Throughout the development of all three user interfaces, I have employed a revision control system, specifically Git, to manage the project's source code efficiently. This system helps me track alterations and uphold a structured and coherent codebase. Through Git, I have the capability to initiate branches for the development of new features or corrections, seamlessly merge modifications, and revert changes when necessary.

Addressing the risk of hardware failure, which, although not highly probable, poses a threat of significant data loss and project disruption, I have integrated the use of GitLab which serves as a version control repository hosted online, ensuring the safe storage of all project code and data remotely. By committing code regularly to GitLab, I safeguard my work against local hardware malfunctions, guaranteeing that no essential data is forfeited. This approach not only secures critical project data but also facilitates the management of different program versions. It allows for the retrieval of previous code versions should any issues arise, thereby reducing potential time loss and effort. This strategy endures continuous progress and resilience throughout the project lifecycle.

### 4.2 Test-Driven Development (TDD)

Throughout the development of my Java Swing application, I've adhered to a Test-Driven Development (TDD) approach. I have been using Junit 5, for testing the Swing applications actual GUI, I will be using a dedicated GUI testing library. One popular library for Swing GUI testing is AssertJ Swing. This methodology has been instrumental in guiding the design and implementation of my project, ensuring that each functionality is robust, meets the requirements, and remains bug-free as new features are added, or existing ones are modified. Here's how I've applied TDD in the development process:

For each new feature or functionality, I start by writing tests that define the expected behaviour. These tests initially fail, as the feature hasn't been implemented yet. This step helps clarify what I'm about to implement and sets a clear goal for the functionality. For instance, when I added the ability to add notes to the application, I first wrote tests to ensure that notes could be added, listed, and persisted correctly. The next step is to write the minimum amount of code required to make these tests pass. For the note’s functionality, this meant implementing the methods to add and retrieve notes. My priority was to get a green light from the tests, indicating that the basic requirements were met. Then I move on to refactoring the code. This involves cleaning up, optimizing, and applying design patterns where appropriate, all the while ensuring that the tests continue to pass. This step is crucial for maintaining code quality and readability.

TDD is inherently iterative, after refactoring, I do all of this again for whatever I implement next. This makes sure all of my code has been thoroughly tested and it also facilitates regression testing, as all tests are rerun after new changes to catch any potential side effects early. Another reason I am using TDD is that the test cases serve as executable documentation, and they provide examples of how your code is expected to behave which make it easier for others to understand my code.

However, TDD also poses challenges, especially around writing effective tests and managing the time investment required upfront. It has been very hard testing Java Swing components especially, for example some of the problems I had were due to the FrameFixture class's limitations in directly accessing the menu bar for automated GUI testing. I overcame this obstacle by employing the robot () method from AssertJ Swing, which provided access to the Robot instance, enabling low-level interactions with Swing components and facilitating comprehensive testing. Balancing the depth and breadth of tests has been a learning curve, requiring a thoughtful approach to ensure that tests are meaningful and cover the expected behaviour without becoming a maintenance burden.

A screenshot of a computer

Description automatically generatedA white background with text

Description automatically generated with medium confidence

### 4.3 User-Centred Design (UCD)

UCD focuses on understanding the needs and preferences of users. It involves techniques like personas, user stories, and usability testing to create interfaces that align closely with user expectations. This is extremely relevant to my project as HCI is all about how a user interacts with the system and my UIs are designed around how a user would use them.

The decision to use User-Centred Design (UCD) was driven by its emphasis on understanding and addressing the specific needs, preferences, and contexts of the end-users. Given that Human-Computer Interaction fundamentally revolves around optimizing the interaction between users and systems, UCD was a natural fit for my project. This methodology ensured that the user interfaces (UIs) I designed were not only functional but also intuitive and satisfying for the users to interact with.

By involving users early and throughout the design process, I was able to identify and fix usability issues pre-emptively, leading to interfaces that are easier and more pleasant to use. Tailoring the design to meet user expectations significantly enhanced their satisfaction and engagement with the UIs.

While UCD was well-suited to my project's objectives, there were other methodologies I could have considered:

Goal-Directed Design: This approach also focuses on users but emphasizes achieving specific objectives. It might have been useful if my project had more narrowly defined goals for each interaction.

Activity-Centred Design: ACD focuses more on the tasks that need to be accomplished rather than the users themselves. This could have been an alternative if the primary objective was optimizing task efficiency, regardless of user preferences.

### 4.4 Code Quality Tools

To maintain consistent code styling across my projects, I have integrated Prettier with Visual Studio Code. This tool significantly streamlines the development process by automating code formatting, which not only saves considerable time but also enables me to concentrate more on the logic and functionality of the code rather than its aesthetic layout. Prettier ensures adherence to a unified coding standard across all my user interfaces. Additionally, I have employed SVGs for icons, eliminating the need for external dependencies like Font Awesome, thus simplifying project setup and maintenance.

### 4.5 My process of design:

My design process initiated with identifying the requirements of my target-users, followed by an analysis aimed at determining how to meet those requirements. This led to a cycle of designing prototypes and testing the user interfaces, incorporating feedback after each iteration to refine and enhance the designs. This approach encapsulates my methodical journey in designing my user interfaces.

A diagram of my design process

Description automatically generated

#### 4.5.1 Agile Methodology

Agile methodology was instrumental due to its iterative development cycle and emphasis on user feedback. This was particularly beneficial during the design and prototyping stages of the UIs. As my project involved creating interfaces that catered to different user needs, the Agile approach allowed for continuous improvement through iterative design, development, and testing phases. It promotes, frequent feedback, and iterative development. The feedback gathered from testing sessions led to refinements, enhancing the usability and accessibility of the interfaces. This was evident in the prototyping and user testing phases, where changes were made based on actual user interactions with the UIs.

While Agile was the backbone of the project, certain aspects of the Waterfall methodology were also integrated, particularly in the initial planning and design stages. The systematic and sequential nature of the Waterfall model was useful for the comprehensive literature review and the initial detailed project planning. This ensured a strong foundation and clear direction and helped when defining the scope and objectives clearly, which is a strength of the Waterfall model.

The decision against using other frameworks was driven by the nature of UI design and development, which demands flexibility and responsiveness to user input. For example, the Waterfall model's linear and sequential phases would have limited the ability to make quick adjustments based on user feedback or to revisit earlier stages without significant time and resource implications.

### 4.6 Design patterns

My Java project incorporates several design patterns. Design patterns are typical solutions to common problems in software design, offering standardized approaches to organize objects and interactions. Here are the design patterns I have used:

Creational – Singleton, Factory

Behavioural – State

Architectural - MVC

#### Model-View-Controller (MVC)

Purpose: Separates the application's concerns into three main components: the model (data), the view (user interface), and the controller (business logic).

Usage: Your application clearly follows this pattern. NotesModel manages the application data, NotesView handles the display and user interaction, and NotesController bridges the two by updating the model in response to user actions and updating the view when the model changes.

How Used:

Model (NotesModel): Manages the application's data and state, including note contents. It implements the Subject interface to notify observers about changes in the notes data.

View (NotesView): Presents the notes and other UI components to the user. It implements the Observer interface, allowing it to react to changes in the NotesModel and update the UI accordingly.

Controller (NotesController): Acts as an intermediary between the NotesModel and NotesView. It processes user actions (e.g., adding a note through the view) and updates the model. It also updates the view in response to model changes.

#### Singleton Pattern

The use of classes like JFrame, where only one instance of the main application window `frame` exists throughout the application. The Singleton pattern ensures a class has only one instance and provides a global point of access to it.

Purpose: Ensures a class has only one instance and provides a global point of access to it.

Usage: While not directly shown in the provided classes, applications often use the Singleton pattern for managing the controller or model instances, especially if they need to be accessed globally from different parts of the application.

#### Command Pattern

The command pattern is used to encapsulate a request as an object, my action listeners, particularly those that perform file operations (‘openFile’, ‘saveFile’, etc.) and editing actions (‘undo’, ‘redo’), implicitly follow the Command pattern by encapsulating the action logic within the listener method.

In FileMenuHandler.java, the actions for menu items are encapsulated in ActionListener anonymous classes, which is a form of the Command pattern.

A black background with red and blue text

Description automatically generated

Purpose: Encapsulates a request as an object, thereby allowing users to parameterize clients with queues, requests, and operations.

Usage: This pattern isn't explicitly implemented in the provided code, but actions in MVC (e.g., button clicks in the view that trigger methods in the controller) can be considered a form of the Command pattern, where user actions are encapsulated as commands that the controller executes.

#### Observer Pattern

The Observer pattern is used when there is one-to-many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically. The Swing event handling mechanism itself is an example of the Observer pattern, where the UI elements (observers) listen to and react to events (subjects).

Listeners attached to UI components, like buttons and menus, act as observers to user actions (events).

A computer code on a black background

Description automatically generated

Purpose: Allows objects to notify other objects about changes in their state. The observer pattern is crucial for implementing event handling systems.

Usage: In your application, NotesModel implements the Subject interface, and NotesView implements the Observer interface. This setup enables the model to notify the view(s) whenever there are changes in the data (e.g., a note is added or removed), allowing the view to update itself accordingly.

How Used:

The application implements a custom Observer pattern with the Subject and Observer interfaces.

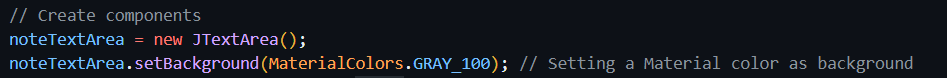
Subject (NotesModel): Holds a list of observers (NotesView) and notifies them of changes by calling their update() method.

Observer (NotesView): Implements the update() method to refresh the UI when notified by the model. This mechanism ensures that the view reflects the current state of the model, allowing for dynamic updates.

#### Decorator Pattern

The Decorator pattern is used to add new functionalities to objects dynamically without altering their structure. This pattern creates a decorator class which wraps the original class and provides additional functionality keeping the class methods signature intact. The use of Swing components with various look-and-feel and utility methods (like MaterialUIMovement.getMovement, setting fonts, and colours) use the Decorator pattern.

Applying material design look-and-feel and movement effects to components:



#### Factory Method Pattern

The Factory Method Pattern defines an interface for creating an object, but let’s subclasses alter the type of objects that will be created. I am using JMenuItem here to create multiple different items.

A screen shot of a computer program

Description automatically generated

#### Java Application UML

# 4.7 Other software engineering

### 4.7.1 Automated testing and CI/CD

In the development of my Java Swing application, I've incorporated automated testing using JUnit 5, which plays a crucial role in my Continuous Integration/Continuous Deployment (CI/CD) pipeline. This integration has significantly streamlined my development process, ensuring that every change pushed to the repository is automatically tested. By ensuring that every component of the application behaves as expected under various conditions, I've minimized bugs and improved the overall robustness of the application. I have done the same for my react application using Jest.

The immediate feedback from the CI pipeline allows me to quickly identify and fix any issues introduced by the latest changes. This practice not only minimizes integration problems but also ensures that the application remains in a stable and releasable state after each commit. By integrating and testing changes frequently, I've dramatically reduced integration issues, making the development process smoother and more predictable. Continuous testing ensures that defects are caught and addressed early, resulting in a higher quality product.

#### React Developer tools testing

In the process of enhancing our React application, we strategically utilized React Developer Tools to conduct thorough testing and performance optimization. This indispensable tool enabled us to delve into our application's React component tree, providing deep insights into the current props, state, and performance metrics of our components in real-time. By leveraging the Profiler feature within React Developer Tools, we meticulously recorded and analyzed performance profiles during key user interactions and scenarios. This approach allowed us to identify performance bottlenecks and components that were rendering unnecessarily or taking longer than expected to render.

We created several testing profiles focused on different aspects of the application. For instance, one profile concentrated on the initial load time and the efficiency of lazy-loaded components, highlighting opportunities to reduce bundle sizes and optimize resource loading. Another profile examined the responsiveness and re-rendering behavior of our dynamic components, such as forms and interactive charts, under various user interactions. Through these profiles, we were able to pinpoint specific areas where performance optimizations could be applied, such as memoizing components to prevent unnecessary re-renders and optimizing state updates to enhance user experience. The documentation of these profiles, complete with detailed findings and screenshots from React Developer Tools, serves as a comprehensive guide for our ongoing optimization efforts, ensuring our application delivers a fast, responsive, and efficient user experience.

#### LightHouse testing

Lighthouse is an open-source, automated tool for improving the quality of web pages. It can be run against any web page, public or requiring authentication. Lighthouse audits for performance, accessibility, progressive web apps (PWAs), SEO, and best practices. It provides a report on how well the page did and offers suggestions on how to improve it. The tool is available in Chrome's Developer Tools, among other formats, making it easily accessible for developers and webmasters.

The document you've shared, such as the PDF generated from a Lighthouse test, provides a comprehensive overview of various metrics and diagnostics relevant to the performance and quality of a webpage. For instance, it details metrics like First Contentful Paint, Speed Index, and Largest Contentful Paint, which are crucial for understanding the page's load time and visual stability. Additionally, it offers insights into potential improvements, such as enabling text compression, reducing unused JavaScript, and minifying JavaScript, each with estimated savings on load times or resource sizes.

Lighthouse's testing is fundamentally different from what React Developer Tools offer. While React Developer Tools focus on the development and debugging of React components, providing insights into component props, state, and React-specific performance optimizations, Lighthouse provides a broader overview of a webpage's performance and quality from a holistic standpoint. It encompasses not just the React elements but all resources, scripting, accessibility, and best practices that affect the page's overall performance and user experience.

The Lighthouse report you've generated includes detailed diagnostics and suggestions for improvement across multiple areas, including performance, accessibility, best practices, SEO, and PWA compliance. This comprehensive feedback loop is instrumental in pinpointing areas for optimization, guiding developers through the process of enhancing the webpage's efficiency, accessibility, and user engagement. Unlike the component-specific insights from React Developer Tools, Lighthouse's reports offer a high-level assessment of the webpage's readiness and performance, making it an essential tool for web development and optimization.

### 4.7.2 Refactoring

This ongoing process of refactoring helps in keeping the codebase clean and avoiding the accumulation of code smells. Whether it's simplifying complex logic, renaming variables for clarity, or removing redundant code, these efforts contribute significantly to the overall health of the project. I strive to follow clean code principles, making sure that classes and methods are focused on a single responsibility and that names are descriptive and intuitive. Refactoring with these objectives in mind makes the code not only easier for me to work with but also for others to understand.

### 4.7.3 Documentation

Throughout the development of my Java Swing application, I've placed a strong emphasis on comprehensive documentation, fully aware of its importance in ensuring maintainability and facilitating ease of use. I have been using JavaDoc for my note’s application and JSDoc for my react app. These both automatically generate a website documenting my code. UML diagram

By embedding documentation directly in the code through Javadoc comments, I've ensured that the documentation stays relevant and up to date with the code it describes. This approach has not only streamlined the process of writing and maintaining documentation but has also made it accessible to anyone looking into the code.

When writing Javadoc comments, I've adhered to a structured format that outlines the purpose of each class, method, and its parameters, return values, and any exceptions that can be thrown. This structured approach aids in clarity, making it easier for developers to understand the functionality and purpose of different parts of the application at a glance. This is also the same structure for JSDoc. Tree of react components

Shopping website- Directory structure graph explain what each directory and file does and comments on all HTML CSS JS components

A black background with white text

Description automatically generated

I have also created documentation for each of my UIs designed to provide clear instructions and examples that help users and developers make the most out of the application. I’ve created these as separate guides.

#### Coding standards for java:

Adhering to standard naming conventions makes it easier for other developers to read and understand the code. By using a consistent naming scheme, developers can avoid naming conflicts and make it easier to find and fix errors. The type of fault this standard avoids is naming errors, where variables, methods or classes have names that don't accurately represent their purpose or are difficult to read.

exception handling is an important feature of Java that allows developers to gracefully handle errors that may occur during runtime. By using try-catch blocks, developers can catch and handle errors in a controlled manner, rather than allowing them to cause a program to crash. The type of fault this standard avoids is runtime errors that may cause the program to terminate abruptly or fail to execute a certain function.

Always close resources using try-with-resources. Failure to close resources can lead to memory leaks, slow performance, and other issues. The type of fault this standard avoids is resource management errors, where resources are not properly managed, leading to memory leaks, degraded performance, or other issues. A screenshot of a computer code

Description automatically generated

5. Technical Decisions

# 5.1 Technologies I have used and why:

### 5.1.1 Java Swing:

Swing is part of the Java Foundation Classes and is included in the Java Development Kit (JDK) since version 1.2. It is a GUI widget toolkit that provides a collection of standard graphical components for creating user interfaces (UIs).

Key features of Swing:

Platform independence: Swing components are written entirely in Java and do not rely on any platform-specific code. This means that Swing applications can be run on any computer that has a Java Runtime Environment (JRE).

Extensive set of components: Swing provides a rich set of standard graphical components, including buttons, labels, text fields, menus, and dialog boxes.

I am making the notes application in swing because it will mainly be used on desktop computer and swing is tailored for making desktop applications. To make sure my application is visually appealing I have used material UI swing to help style the components.

The decision to utilize Swing for creating the graphical user interface (GUI) instead of JavaFX was mainly due to its ease of use and faster learning curve, making it a more straightforward choice for quick development. Additionally Swing presents itself as a more stable option, with JavaFX potentially being phased out or replaced by newer technologies [The Battle of JavaFX Vs Java Swing: Which One Is Best For You? (devdojo.com)](https://devdojo.com/anvesh_kumar/the-battle-of-javafx-vs-java-swing-which-one-is-best-for-you#:~:text=The%20choice%20between%20JavaFX%20and%20Java%20Swing%20largely,and%20feel%2C%20then%20Swing%20is%20a%20great%20choice.). Another critical advantage is Swing's inclusion in the standard Java Development Kit (JDK), which ensures compatibility and ease of execution on various systems without the need for additional software installations.

While Swing offers a robust platform for developing graphical user interfaces in Java, it does come with certain limitations that impacted my projects development. Notably, Swing's aesthetic flexibility is constrained compared to other technologies like HTML and CSS, resulting in a visual design that doesn’t fully align with my designs. Additionally, swing lacks support for some of the more advanced or specialized functionalities I had planned, for example the customizable menu was extremely hard to implement and is still not as good as I planned it to be, necessitating compromises or alterations to the original design.

OOP

### 5.1.2 HTML/CSS/JS with Bootstrap:

Bootstrap is a free and open-source front-end framework developed and is a powerful and popular toolkit for building responsive and mobile-first web applications. Bootstrap simplifies the process of designing and styling web pages by providing a set of pre-built components, CSS styles, and JavaScript plugins that can be easily integrated into web projects.

Why I used bootstrap and key features:

Bootstrap provides a pre-built collection of CSS classes and JavaScript components that can significantly expedite the development process which facilitates rapid prototyping.

Bootstrap is built with responsive design principles, ensuring that my website seamlessly adapts to different screen sizes and devices, including desktops, tablets, and smartphones. This is crucial for providing an optimal user experience across various platforms.

Bootstrap's grid system provides a structured layout framework for organizing my website's content. This makes it much easier to create consistent and visually appealing layouts without having to manually adjust margins and padding for each element. Grids help to create a consistent visual experience for users (4).

Grids can also help to make designs more accessible to users with disabilities. For example, users with low vision may find it easier to scan and understand a grid-based layout. Additionally, grids can be used to create designs that are compatible with screen readers and other assistive technologies.

Bootstrap is extensively tested across various browsers to ensure consistent rendering and behaviour across different platforms.

Bootstrap's consistent design language ensures that my website maintains a cohesive look and feel throughout its pages (consistency). This enhances the overall user experience and makes the website more visually pleasing.

I have also implemented the IntroJS library which is a lightweight library for creating step-by-step interactive walkthroughs and guided tours for websites and applications, enhancing user onboarding and feature discovery.

#### These are the reasons why I only used React for my database interface website only and not the shopping website UI:

Simplicity: I only need a simple shopping website with no complex functionality, so using a framework can add unnecessary complexity. Frameworks often come with a lot of code and features that I wouldn’t need and could make my code more difficult to manage and debug. Additionally, frameworks can add additional processing overhead, which can slow down the website.

Control: I need complete control over the look and feel of the website and using a framework can be restrictive. Frameworks often have their own built-in styles and layouts, which can limit the ability to create a unique and personalized design. Additionally, frameworks can make it more difficult to customize the behaviour of the website.

Flexibility: This shopping website will be tested by users many times and I am focusing on having good visuals inn that one so it will be changed many times in the prototyping process as I am using a prototyping approach therefore using a framework can make this a lot more difficult and time consuming.

### 5.1.3 React

Creating a database interface website for business owners and managers with Human-Computer Interaction (HCI) principles in mind is essential for ensuring the platform is user-friendly, efficient, and effective. React, a popular JavaScript library for building user interfaces, is particularly well-suited for this task.

React's component-based architecture enables the construction of encapsulated components that manage their own state, which can then be composed to craft complex UIs. This modular approach is helps achieve greater design coherence, simplifying maintenance, and creating more intuitive interfaces. Such attributes are fundamental for adhering to HCI principles, ensuring that the interface is accessible and understandable for users.

The declarative nature of React simplifies the process of developing interactive UIs. By allowing developers to specify what they aim to achieve (the desired UI state) and letting React handle the efficient updating and rendering of components as data changes, this approach yields code that is more predictable and easier to debug. This is particularly advantageous in HCI, where the goal is to design interfaces that are both intuitive and user-friendly.

Additionally, React benefits from a Rich Ecosystem and Tools, bolstered by a wide array of libraries and tools that enhance both the UX and UI design. For example, I have used the react router library to connect all of my pages together, these resources speed up the development process while ensuring a cohesive and engaging user experience.

Since react is so popular, the community offers a plethora of resources, including tutorials, forums, and third-party tools, all aimed at addressing common design and development hurdles. Access to such a broad support network is instrumental when crafting interfaces that prioritize the user's needs and experiences.

6. Final Deliverables

# 6.1 My UIs - Designs and HCI principles

I will now discuss my UI’s and how I have tailored them to HCI standards and principles, refer to theory section for explanations of the principles that I am mentioning

* I have discussed my code in my documentation and, for example in the CSS files I have described the HCI principles involved

# A screenshot of a computer Description automatically generatedInventory management system:

This interface will have learnability integrated into a help system which will come up with tool tips and guidance on how to use the system. Will also have help and manual section.

To promote efficiency, productivity, and flexibility, all the main functions and options are on the home screen and easily accessible. There will also be shortcuts for the most used functions.

This should also be a memorable interface as it will always have the same layout on every page and on every device, it is used on.

Security and privacy: Provide real-time feedback on inventory changes, updates, and system status. Timely notifications and alerts can help business owners stay informed about critical events affecting their inventory.

There is a + - icon to increase and decrease the size of the Screen, these will have tooltips or name tags to explain what they do.

To facilitate user control and freedom, there will be the options to undo/redo, delete, exit anything they have done.

It has a very minimalist design reducing cognitive load.

Will also have customizable home page as the system will be able to hold a lot of data so the user can put their most important data in the homepage.

To show the systems tatus I will have a loading bar, so the user does not get frustrated thinking nothing is going on.

A screenshot of a computer

Description automatically generated

A screenshot of a login form

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generatedA screenshot of a login page

Description automatically generated

# Notes application:

A computer screen with text and a window

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A black text with a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

A computer keyboard with a keyboard and a keyboard

Description automatically generated with medium confidence

Design is easy on the eyes and simple so students will pay attention and not get bored quickly (satisfaction).

This will also give feedback about the storage of the system and feedback about any changes and actions made e.g., confirmation of change popup.

The help section adds more usability to the application, The notepad application will have an integrated search feature to help users quickly locate specific notes and any functions they need.

Error correction: One way this will be done is Spell checking: The notepad application will include spell checking to help users catch and correct mistakes.

It is also robust as it provides many options and functions for the user.

All components will have consistent names with other word processors to make them easy to recognize.

It will also prevent errors, e.g., data loss from the computer closing abruptly, bad input.

I will also have a language setting for different languages. Also has very simple and neutral language to be easily used by different user groups (internationalization).

Shopping website

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a checkout form

Description automatically generated

A screenshot of a computer

Description automatically generated

This website provides clarity as it is simple and easy to understand, it also has a consistent styling for each page all this also helps with recognition rather than recall. Also aligns with users’ mental models of other websites.

I will provide clear and concise feedback for every action the user does e.g. popup when something is added to the basket.

It will also have audio feedback such as button clicks and error sounds for visually impaired people.

It will have error prevention by making sure that the option a user is doing is what they want to do, e.g., buying something, before a user pays for their items, they will have to check the basket where they can remove items if needed.

I will be providing accessibility by designing the website in a way where it is compatible with devices such as screen readers.

Learnability is also included as it is very simple to use and therefore quickly learnt.

There will also be a bar and a section of the page to show how far the user is in their buying process. User feedback

The categories of items will be like items in other shops.

It has a very minimalist design. (Clarity)

The size of each element indicates its hierarchy.

I created this using grid which can help to make designs more accessible to users with disabilities. For example, users with low vision may find it easier to scan and understand a grid-based layout. Additionally, grids are used for designs that are compatible with screen readers and other assistive technologies

### 6.2 Other HCI methodologies I have included in my designs:

Consistent Names, Clear Choices, ensuring that users encounter uniform and comprehensible terminology throughout the interfaces. This not only simplifies navigation but also fosters a sense of familiarity.

The principle of Recognition over Recall has been prioritized, emphasizing the presentation of recognizable elements over expecting users to recall information. Aiming for a user-friendly experience, I incorporated Aesthetic and Minimalist Designs to eliminate unnecessary clutter and enhance visual appeal. Additionally, strategic use of whitespace has been implemented to convey grouping and improve overall readability.

Size contrast and hierarchy have been effectively employed using varying text sizes and thickness, promoting an intuitive understanding of content structure. Furthermore, I selected a font with a high X-height, as research indicates that a higher X-height facilitates easier online reading for a broader audience.

In adhering to best practices, mixed typed case has been favoured over all caps, allowing for greater vertical variation and providing the reader's eye with more information during the reading process. Initially, I utilized tools like scale and layout to distinguish elements on the page, foregoing heavy reliance on colour.

The incorporation of grids has played a crucial role in establishing a consistent visual experience for users. This design choice ensures a structured and organized layout, contributing to a seamless user interface.

Generic icons have been strategically employed to facilitate repeat recognition, leveraging users' familiarity with visual representations rather than relying solely on textual cues. Furthermore, I have prioritized the design for "glanceability," enabling users to quickly absorb and comprehend essential information.

To draw attention to critical information, such as error messages, I implemented dynamic elements such as flashing or moving features. This approach effectively captures the user's attention and communicates the urgency of the message.

Menus and toolbars/navbars have been thoughtfully designed to minimize user demands, relying on recognition rather than recall. Logical grouping and meaningful naming of menu options provide users with cues for locating the required functions. In toolbars/navbars, icons have been strategically placed to align with users' natural scanning patterns, reducing the need for extensive reading or referencing manuals.

Dialog boxes have been strategically employed for presenting important information, ensuring that users receive essential notifications or prompts. This approach allows for a focused and unambiguous communication channel, enhancing the overall user experience.

7. Professional issues

Professionalism in the field of computing encompasses a broad understanding of the societal implications of technology, as well as a commitment to ethical practices and policies. This commitment is vital in ensuring that technological advancements contribute positively to society, respecting the rights of all individuals affected by their deployment. Professional bodies like the British Computer Society (BCS) and the Association for Computing Machinery (ACM) play a crucial role in upholding these standards, offering guidelines and codes of conduct that emphasize competence, integrity, and the prioritization of public interest.

### 7.1 Correct Citation and Licensing

Ethical and legal considerations require that software developers and researchers properly cite the sources of ideas, code, or other resources they utilize. Moreover, understanding and adhering to software licensing agreements is crucial to respect the intellectual property rights of others and avoid legal complications. Every time I have used code, I had not written myself, I cited where I got it from in the comments of my code.

### 7.2 Accessibility and Usability

Creating technology that is accessible to all, including those with disabilities, is both an ethical obligation and a professional responsibility. The design of user interfaces and systems must prioritize usability to ensure that technologies enhance human capabilities rather than detract from them. It ensures that the benefits of computing technologies are available to a broader audience, promoting inclusivity and equality. My UIs are designed with accessibility in mind due to their HCI considerations.

#### Example from the Public Domain:

A notable example of the repercussions when professional issues are not adequately addressed is the controversy surrounding Cambridge Analytica's use of Facebook data. This case highlighted the importance of ethical data practices, privacy considerations, and the need for transparency in how user data is collected, used, and shared. The backlash from this incident emphasized the societal demand for higher ethical standards in computing and data management.

### 7.3 Professional Issue of Concern in My Project:

In my project, accessibility was a significant professional issue. This concern goes beyond compliance with legal standards like WCAG, it touches on the broader ethical commitment to inclusivity. Making my interfaces accessible involved considering diverse user needs, such as providing alternative text for images (for screen readers), ensuring keyboard navigability, and designing with colour contrasts that accommodate users with visual impairments. The process underscored the ethical dimension of design choices and the professional responsibility to create technology that empowers all users.

8. Project Analysis

### 8.1 Timeline

**Term 1**

Week 1-2 starting 18/09/23:

Start research, [**Human-Computer Interaction (3rd Edition)**](http://www.amazon.com/gp/product/0130461091?ie=UTF8&tag=hci01-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0130461091)**, Alan Dix, Gregory D Abowd, Janet E Finlay and Russell Beale**

**Read chapter 1, The Human, 2.3 Positioning, pointing and drawing, 2.4 Display devices, 2.10 Summary.**

Work on the project plan, draft should be finished by the 27th of September and sent to supervisor.

-Meeting with my project supervisor on Wednesday the 25th of September

-Set up Eclipse and GitLab

Week 3:

Read Paper: **Sinha, Gaurav, Rahul Shahi, and Mani Shankar. Human computer interaction**

Finish Timeline, Bibliography and risk assessment by the 4th of October, then I will submit on the 5th.

Plan and design all 3 UIs using Canva [10], using [7] as a design guide.

Start creating a database with no information stored first plan database schema.

Alan dix. Chapter 3, 4.3 Summary

Week 4:

Continue Database with MySQL, Begin website interface for DBMS.

Continue [**Human-Computer Interaction (3rd Edition)**](http://www.amazon.com/gp/product/0130461091?ie=UTF8&tag=hci01-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0130461091)**, Part 2: Design processes**

Finish basic website layout.

5.4 User focus, 5.6 Navigation design, 5.7 Screen design and layout, 5.8 Iteration and prototyping.

6.3 Usability engineering, 6.5 Design rationale, 7.2 Principles to support usability, 7.5 Golden rules and heuristics, 7.6 HCI patterns, 7.7 Summary.

Week 5:

Add components/widgets like buttons, labels, text fields, and more to the website window.

Set up event handlers to respond to user interactions (e.g., button clicks) for website.

Begin learning Swing [9] for Java desktop application.

8.5 User interface management systems, 8.6 Summary, 9.6 Summary, 10.2 Universal design principles, 10.3 Multi-modal interaction, 10.4 Designing for diversity.

Finish SQL database, Use spring boot for the backend of the DBMS interface.

Read: Designing for the Web: An Introduction to Human-Computer Interaction by Jakob Nielsen

Week 6:

Begin desktop interface with swing.

10.5 Summary, 11.2 Requirements of user support, 11.3 Approaches to user support, 11.5 Designing user support systems.

Watched and made notes on Scott Klemmers lectures. Design Heuristics parts 1 and 2, Visual Design

Watched and made notes on Scott Klemmers lectures. Typography, Grids and Alignment, Reading and Navigation, Designing Studies

Continue working on backend of website to add functionality.

Start writing interim report.

Continue DBMS interface, Website using html/CSS/js.

Continue [**Human-Computer Interaction (3rd Edition)**](http://www.amazon.com/gp/product/0130461091?ie=UTF8&tag=hci01-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0130461091)**, Part 3: Models and theories**. This talks about a user’s cognitive and problem-solving abilities: 12.7 Summary, 13.4 Summary, 14.6 Summary, 16.2 Dialog design notations, 16.3 Diagrammatic notations, 16.6 Dialog analysis and design, 16.7 Summary.

Interim Report: aims, objectives and literature survey.

week 7 – week 8:

Continue application and database interfaces.

17.5 Summary, 18.5 Summary, 19.8 Summary, 20.4 Information and data visualization, 20.5 Summary, 21.2 Understanding hypertext, 21.7 Summary.

Start presentation: explain the aims and objectives clearly, explain the background/relevance/importance of the project and set it in the wider context, I’ve made a broad description of the project - i.e. how parts of the project fit together to form a coherent whole.

Continue [**Human-Computer Interaction (3rd Edition)**](http://www.amazon.com/gp/product/0130461091?ie=UTF8&tag=hci01-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0130461091)**, Part 4:Outside the box**

Finish Report Draft

Interim Report: summary of completed work.

bibliography and citations.

Background Theory

Software Engineering

Week 9 – Week 10:

Week 9 submit report draft.

Conduct user testing on the three implemented interfaces. (Beginning of Week 9)

Continue presentation and report.

Presentation: briefly explain the theory underpinning the individual parts of the project (for example how algorithms work, or which architectural options existed including their benefits/ drawbacks), defend and justify decisions made during the project.

Interim report due 1st December

Presentation to be submitted 2nd December.

Week 11:

Presentation on 4th December

**Term 2**

Week 1 (Starting 15/1/2024):

Begin on affordance features in for shopping website.

And visual effects for help menu guiding you through the interface.

Put dashboard page of inventory interface into a function for react.

Week 2 – Week 3:

Add functionality to Basket in shopping website.

Begin visuals for voice control of shopping website.

Write new sections for aim and objectives in final report, with the new changes for the interfaces.

Write Project specification section of report – Add new changes (only 2 interfaces)

Also change My UI’s section

Rewrite target Audience section with feedback from interim report.

Put Navbar for shopping website into react component.

Week 4:

Report, Literature Review and Background Reading

Add functionality to voice control in shopping website.

Add customizable menu to notes application.

Write these sections of report.

Technologies I have used – Add React

Add more in-depth information to literature review for

HCI by Alan Dix

Designing the user Interface by Ben Shneiderman

Empirical Research in HCI by Scott MacKenzie

Edit HCI goals section to reflect new plan.

Create draft for My progress section, to be finished when interfaces visuals are complete.

Work completed sub-section.

Process of design sub-section

Designs and HCI principles

Week 5:

Add Internationalization principle for notes application, Menu which can change the language of the application.

Colourful web site with images and navigation

Documentation: Create user documentation and guides to help users understand how to use your application effectively

Report – write software engineering section.

Talk about other methodologies I could have used besides UCD and why.

Also, other methodologies other than Agile and why

Add Loading bar to all pages in shopping website (see system status)

Add more citations to report.

Add dark mode options to notes application.

Week 6:

Continue User Manual for the implemented interfaces.

Add how to run section to appendix.

Conduct user testing for all UIs, plan to implement main feedback.

Complete Self-evaluation section and critical analysis section of report

Discussion of project achievements and over how successful it was.

Complete report (Week 22) and send to supervisor.

Write conclusion.

Create mobile and tablet versions of shopping website with bootstrap.

Use React DevTools extension to test website and debug.

Week 7 – Week 8:

Implement all feedback of report from supervisor.

Conduct 2nd User interface testing for both interfaces.

Using selenium for shopping website

Add output to report appendix.

Make video of UI’s and add YouTube link to report.

Week 9 – Week 10:

22nd March, Final report and programs due

# 8.2 Self Evaluation

Conclusion

In conclusion, the development of three distinct user interfaces, each tailored to cater to specific user groups and tasks, helped me understand the crucial role HCI principles play in creating intuitive, efficient, and engaging digital environments. Through a meticulous design process that incorporated a comprehensive survey of relevant literature, the application of a wide array of HCI methodologies, and the adherence to software engineering principles, this project has demonstrated the impact of user-centred design on the usability and overall user experience of digital interfaces.

Employing technologies such as Java Swing for a notes application, HTML/CSS/JS with Bootstrap for a shopping website, and React for a database interface, each choice was underpinned by a rationale aimed at maximizing functionality, accessibility, and user satisfaction. The challenges encountered, particularly with Swing's limitations in aesthetic flexibility and the complexity of testing GUI components, highlight the iterative nature of design and the necessity of aligning technological choices with HCI principles to overcome obstacles and refine the user experience.

Moreover, the integration of tools like Git for version control and the adoption of Test-Driven Development (TDD) emphasize the importance of maintaining a well-organized, robust codebase and the benefits of continuous testing in the development process. These technical decisions, coupled with a commitment to creating accessible and inclusive interfaces, reflect the project's goal to enhance the interaction between users and computer systems, making technology more approachable and usable.

HCI is most likely to become the only most global research topic of the AI (Artificial Intelligence) research

community. The sudden discovery in HCI design could bring radical change in the world. Many aspects of the HCI

technology, which are concerned with interpretations of human behaviour at deeper level. HCI will bring a massive

change in the world. Since the human computer interaction is based on the interaction of the humans with the computers,

it would be more preferred as it is easy to use and is totally dependent on the humans/users and works on the users

instructions. A small work in this field will ease the work of people in the upcoming time.

Appendix

### Instructions for each UI

### Diary:

Bibliography and citations

[1]. [*Human-Computer Interaction (3rd Edition)*](http://www.amazon.com/gp/product/0130461091?ie=UTF8&tag=hci01-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0130461091)*, Alan Dix, Gregory D Abowd, Janet E Finlay and Russell Beale***,** this book provides a comprehensive overview of the field of HCI.

Resources that have helped me prepare my plan:

[2].[*https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=Types%20of%20user%20interfaces&text=graphical%20user%20interface%20(GUI),touch%20user%20interface*](https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=Types%20of%20user%20interfaces&text=graphical%20user%20interface%20(GUI),touch%20user%20interface). This article helped me decide which Interfaces I was going to make so I could plan how to make them.

[3]. *Sinha, Gaurav, Rahul Shahi, and Mani Shankar. "Human computer interaction." 2010 3rd International Conference on Emerging Trends in Engineering and Technology. IEEE, 2010*. This paper is a basic introduction to HCI and talks about the history and how the field has evolved, and what the future of HCI could be.

[4]. [*https://youtu.be/WW1g3UT2zww?si=lgXaFbYio-kMyJAt*](https://youtu.be/WW1g3UT2zww?si=lgXaFbYio-kMyJAt)

This is a playlist of lectures by Scott Klemmer and is a full course on HCI offered by Stanford University

[5]. [*Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition)*](https://www.amazon.com/Designing-User-Interface-Human-Computer-Interaction/dp/013438038X/)*,*

This is atextbook on user interface (UI) design. create UIs that are easy to use and pleasurable.

[6]. *MacKenzie, I. S. (2012). Human-computer interaction: An empirical research perspective*. This is a book by I. Scott MacKenzie that provides a comprehensive overview of empirical research in human-computer interaction (HCI). The book is divided into two parts:

Part 1: Foundations: This part covers the basics of HCI research, including the history of HCI, the human factor, interaction elements, and the fundamentals of science and research.

Part 2: Methods and Applications: This part covers the different methods used to conduct HCI research, as well as how to apply HCI research findings to the design of new computer systems and interfaces.

[7].[*https://www.creativebloq.com/features/ui-design*](https://www.creativebloq.com/features/ui-design)

This article provides a comprehensive overview of UI design, covering everything from the basics to advanced techniques. I will use this when designing my UI.

[8]. [*https://www.youtube.com/watch?v=He-1O8Pa4SE&list=PLlGZc17KPrVCGRKtgbdvnGshN8AePlqpd*](https://www.youtube.com/watch?v=He-1O8Pa4SE&list=PLlGZc17KPrVCGRKtgbdvnGshN8AePlqpd)

This is a playlist that guides you thorough java swing and how to use it to create a good-looking GUI.

[9].[*https://docs.oracle.com/javase/tutorial/uiswing/*](https://docs.oracle.com/javase/tutorial/uiswing/)*.*

This is the Swing manual and documentation.

[10]. [*https://www.canva.com/*](https://www.canva.com/)

This is a website which I used to design my UI’s.

[11]. [*https://www.smashingmagazine.com/2008/01/10-principles-of-effective-web-design/*](https://www.smashingmagazine.com/2008/01/10-principles-of-effective-web-design/)

[12].[*https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb*](https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb)

[13]*.* [*https://www.w3.org/TR/WCAG21/*](https://www.w3.org/TR/WCAG21/)

[14]. *Norman, D. A. (2004). Emotional design: Why we love (or hate) everyday things. Basic Books/Hachette Book Group.*

This Book discusses how emotion influences design and how different emotions affect users when interacting with interfaces. It also talks about design techniques and practices to help convey emotions when designing interfaces.

[15]. <https://react.dev/>

This is the documentation for react.

[16]. Bansal, Himanshu & Khan, Rizwan. (2018). A Review Paper on Human Computer Interaction. International Journal of Advanced Research in Computer Science and Software Engineering. 8. 53. 10.23956/ijarcsse. v8i4.630.

[17] <https://getbootstrap.com/>

This is the bootstrap documentation.

[18] <https://introjs.com/>

This is the inrojs documentation.

[19] <https://material-ui-swing.github.io/docs/>

This is the material Ui documentation.

[20]<https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction>

[21] <https://pooyasetayesh.medium.com/human-computer-interaction-hci-bridging-the-gap-between-people-and-technology-df5e07333077>

[22] Krug, S. (2006), Don’t Make Me Think: A Common-Sense Approach to Web Usability, Berkeley, CA: New Riders

[23]