Developing an Ethics Case Study Repository for Effective Teaching and Learning: Ethics CS Portal

Oscar Alberigo

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2023

Abstract

Inspired by the growing importance for computer scientists and software engineers to think more ethically about their implementations and modern technology in general I have developed and designed the Ethics CS Portal. This tool was devised to support the teaching of computer ethics across higher education which aims to provide a platform where university lecturers can easily access a wide range of case studies relating to computer ethics to teach in their courses. The aim is that this tool will reduce the time spent by lecturers sourcing relevant teaching materials and therefore support the integration of more ethical discussions in technical courses, leading to a more socially and ethically aware population of programmers.

The tool's features and implementation is backed by previous research on computer ethics pedagogy [21] along with research carried out in the development of the tool which consisted of questionnaires, heuristic walkthroughs, and user acceptance testing.

The first stage of design was to gather requirements from participants to identify the key features of the tool as envisioned by them. This was followed by several iterations of the tool's design, gathering feedback from participants with experience in human-computer interaction along the way. Finally, the tool was tested on the participants to see if it met the requirements set out in the first stage.

The output from this work is a website that can support lecturers in finding, storing, and adding case studies to the database. The final summative feedback from the participants concluded that while there is potential for future development to add features and improve the usability of the website, the final product is a tool containing real-life case studies in ethics with different applications.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics Committee.
Ethics application number: RT #7307
Date when approval was obtained: 2023-02-21
Date when second approval was obtained: 2022-03-07
The participants' information sheets and consent forms are included in appendix D.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Oscar Alberigo*)

Acknowledgements

Firstly, I would like to thank all the people that participated in the studies, their feedback was very insightful and helped shaped the design and features of the final product.

I would also like to thank my supervisor James Garforth for being so invested in my project. Our long chats discussing the purpose and direction of the tool, especially at the beginning of the project were incredibly helpful. Also, for his constant willingness to answer questions and organise meetings, even throughout the busiest times of the year.

Table of Contents

Chapter 1

Introduction

1.1 Motivation and context

The field of Computer Science (CS) is currently a topic of intense debate and discussion. It is not uncommon for CS to make headlines, but unfortunately, these headlines often highlight negative impacts that technology has had on society, such as data leaks, algorithmic bias, and unwanted surveillance.

Articles of this nature also highlight the ever-increasing influence that technology has on our daily lives, as it becomes more deeply integrated into the way we interact, conduct business, and plan our day.

As a result, developers have the ability to shape the modern world as we move even further into this technological era. With this great power comes great responsibility, and it is the responsibility of developers and others in this field to not only make their technology effective and efficient but also just and fair. The harm caused by unjust and unfair technology can be significant, both for individuals and companies, as seen in numerous press scandals [1] [9].

Developers acquire their technological skills in various ways, but one common path is through university degree programmes. At the end of the degree, it is not only important for a student to display a required understanding of the main knowledge in their field but also “perform work within a professional and ethical framework”[20].

1.2 Focus and scope

This report in particular will focus on helping achieve the latter of the previous statements, namely, improving a CS student’s ability to consider the social and ethical implications of their technology. The Ethics CS Portal will be developed with a case study-based approach as it has been identified as an effective way to teach ethics and promote moral reasoning [11] [14].

University plays an important role in preparing students for the world of work [8]. To allow lecturers to integrate computer ethics into their courses and the CS curriculum in general it is important that collating course material be as uncomplicated as possible. The Ethics CS Portal was designed with this in mind, to provide lecturers with a repository of case studies in one place for effortless browsing.

1.3 Relevance and importance

In the past, there has been a perception that incorporating computer ethics into CS degrees is burdensome [15]. This article states that CS professors simply do not have the knowledge or experience to teach ethics material. Another study [24] highlights the fact that departments just don't have enough resources, time, and staff to be able to teach computer ethics to students, noting however, that they still view this field as 'important'. The task for the proposed tool is thus clear - to allow more universities to integrate computer ethics into their CS degrees with ease. The tool must therefore simplify the process of finding case studies in the hope that it will ease the workload on lecturers when creating teaching material, enabling universities to incorporate computer ethics in their programmes.

At the time of this report's publication, the tool is available as a fully-functional website, accessible here. The tool serves as a repository of case studies that illustrate issues that arise when technology is developed without considering its potential ethical consequences. The website also operates as a community-driven database, enabling lecturers to contribute case studies to ensure the database remains current on emerging ethical issues.

1.4 Research questions

The main aim of this project is to develop a tool that will assist lecturers in selecting case studies for computer ethics modules/courses. In order to optimise the tool's efficacy, this report seeks to answer the following questions:

1. How have universities regarded the integration of computer ethics into CS degrees over the years?
2. What are the most effective methods of teaching computer ethics?
3. What are the most effective methods of requirements gathering and design that could be used during the implementation of the proposed tool?
4. Are there similar tools in this field and what are their strengths and weaknesses?
5. What are the requirements of the tool?
6. How is the design perceived by human-computer interaction (HCI) students?
7. Is the final product a viable tool? Is it usable? And does it meet the initial requirements?

1.5 Contributions

This section highlights the main outcomes of the research and studies carried out in this report.

- Computer ethics has been an equivocal topic for scholars in the past but the integration of ethics in CS degrees is increasing.
- The case study method has been established as an effective pedagogical method for delivering computer ethics.
- Other tools exist on the web but are either lacking features to simplify the process of finding case studies or their databases require too much time and effort to populate and might not be able to keep up with fast-changing technology issues.
- Requirements identified by participants noted that the tool should be community driven, have a quick easy-to-add approach, and have a way to store/bookmark case studies.
- The tool was firstly sketched on paper, then designed on Proto.io and finally coded in React.
- Some usability bugs were found during the heuristic walkthrough, for example, difficulties adding a case study and the naming of certain tabs. These were amended before the final evaluation.
- The overall evaluation from participants was positive with only a few test cases failing. These were amended before the launch of the final product.

Chapter 2

Background reading

This chapter will outline the background reading which was carried out prior to commencing any design or implementation in order to construct a solid foundation for the development of the tool. It reviews universities' changing views on computer ethics over the years as well as the current methods for teaching ethics in higher education. Moreover, it investigates the current tools for aiding the teaching of computer ethics and the importance of including computer ethics in higher education in the modern world. Lastly, human-computer interaction (HCI) techniques are discussed to provide an overview of the methodology used in creating the Ethics CS Portal.

2.1 University views on teaching ethics in the past

Integrating ethics into the Computer Science (CS) curriculum in the past has been an equivocal topic. Some scholars were opposed to the integration while others believed it would be beneficial for students. During the 1980s scholars who were opposed to teaching ethics argued that the curriculum was already crowded as it was and hence to spend time illuminating the societal impacts of technology would diminish the students' learning of technical information [15]. Miller on the other hand states that one need not think of teaching ethics in CS as withdrawing from students learning but that ethics and theoretical concepts go hand in hand and can be integrated harmoniously into a course. Miller [15] also states that to better comprehend technical issues and teach them effectively, it is important to consider their social context. Similarly, understanding the societal aspects of computing is best achieved by delving into the underlying technical details. Furthermore, scholars have previously disputed the fact that CS professors are not comfortable teaching ethics and therefore the ethics that will be taught would be "diluted at best and possibly erroneous" [15]. A counterargument put forward by Johnson discusses the notion that regular CS professors should be the ones teaching computer ethics "to emphasize to students that social impact issues are a fundamental part of computer science, not some tangential topic that they take somewhere else" [10].

Fast forward to the very start of the 21st Century and the shift in mindset is clear.

University course accreditation companies like ABET [1] in America begin to set a minimum requirement of teaching ethics in CS courses. Universities start adopting many different techniques for teaching ethics in computer science, Michael J. Quinn's paper [21] shows the results from a survey across 50 departments in America with accredited computer science programs. The results showed that fifty-five percent of departments taught ethics as a stand-alone ethics course within the CS department. Thirty percent of universities incorporate discussions of social and ethical issues of computing within other computer science courses. This leaves fifteen percent of universities that require students to take a separate ethics course taught by another department, typically philosophy. The rise in standalone ethics in CS courses across the board is implemented to solve a pragmatic problem. The ABET accreditation agency can more easily approve standard IV-17 which states that "There must be sufficient coverage of social and ethical implications of computing to give students an understanding of a broad range of issues in this area" [2] if the university has standalone computer ethics courses. Despite the rise in ethics courses in universities, the motivation seems to be forced by the accreditation company rather than improving students' knowledge of the subject. Within the UK Higher Education system the importance of ethics in CS courses is highlighted in the QAA Subject Benchmark Statement for Computing [20], which states that university providers might need to incorporate a wider range of materials and resources, including ethics to "fully capture the specific character of their particular degree course".

2.2 Current views on ethics in CS

2.2.1 The changing school curricula

Computer Science is currently undergoing an immense change in the school curriculum [19]. Nowadays the term 'ICT' is used less in primary and secondary schools across the UK and is being replaced by 'CS'. The idea behind 'ICT' was to teach the skills needed to use existing technology, namely office applications and the web in general. The new goal of teaching CS lies in improving pupils' cognitive thinking, programming, problem-solving, and creative skills. This change in the curriculum means computer ethics will become an essential topic for learners of all ages [24].

2.2.2 University views on computer ethics nowadays

Currently, it is becoming more commonplace for institutions to teach computer ethics to students at various occasions during their degree. The aim is to create a generation of programmers who not only build efficient and useful programs but also consider the social and ethical consequences of their implementations. In a recent paper [24], universities around Europe were asked whether they teach computer ethics in their university. Of the 62 universities, 63% of them taught computer ethics somewhere in the University. In this study, it was also concluded that most computer ethics is taught as a

[1] https://www.abet.org/accreditation/

[2] https://www.uv.es/alfa-acro/documentos/documentosinteres/36.pdf

standalone course in a CS degree. The report also discusses the widespread accordance regarding the significance of teaching computer ethics [24]. It is clear that there has been a shift in thinking since the 1980s as technology has become more integrated into our daily lives. Interestingly, the results from the same paper show that a third of the universities do not teach computer ethics due to a lack of staff expertise, lack of staff availability, and lack of time. This boils down to the debate mentioned previously over who should teach computer ethics. Finally, the results from Fielser's [4] study noted that there has recently been a slight shift from standalone ethics courses to integrating computer ethics into different technical courses across a CS degree.

2.3 Current computer ethics pedagogy

The pedagogy for computer ethics differs from that of technical concepts. Quinn's paper [21] discusses the most effective methods for teaching computer ethics. The research found that since the field is a very open-ended subject, ethics is best taught in a discussion-based approach. This falls into the category of "Learning through discussion" of the conversational framework of teaching [3]. One of the ways these discussions could be guided is by the lecturer, the discussions expose the students to points of view that may be different from their own but allow them to listen to arguments and form them themselves. Having to think about a well-formed argument ensures the student slows down and sees both sides of the coin. Role-playing along with ensuring that the topics being covered are relevant and thought-provoking has also been deemed an effective way to increase student engagement. Group discussions are an effective way to have productive conversations going on simultaneously and by presenting their case to the rest of the class, students improve their presentation skills along with their ability to think on their feet. These discussions are usually fueled by case studies. Case studies have been proven to be an effective method to teach ethical concepts to students across different fields [14] [11]. In particular, *Francis, et al.,* [14] highlight that the teaching of scientific integrity in nursing and biomedical sciences "is greatly facilitated by a case study approach." and stresses that case studies present real-life issues that students must discuss and solve by using a mixture of their own personal values and acceptable scientific standards in their field of study. This article also notes that the effectiveness of using case studies as a means to educate students on ethical concepts can be expanded to "other topics appropriate for scientific integrity". More evidence was found, this time in the field of computer science by Miller [15] which states that by using the case study approach, ethical reasoning "can be naturally incorporated into existing lectures and used with existing textbooks" [15]. Another approach that was trialed by Don Gotterbarn [5] was a software engineering method that combines computer ethics and a software engineering project late in the student's college career. According to Gotterbarn, this is a more practical and effective way to teach ethics in CS as the students are encouraged to make good ethical decisions whilst implementing a piece of software which is something that one hopes they will carry on into their professional life. These methods, along with writing assignments, short-weekly quizzes, and essay questions are the standard procedures taught across the board in an ethics environment.

[3]- https://abc-ld.org/6-learning-types/

2.4 Other tools in this field

In developing the Ethics CS Portal it was important to research other tools with similar use cases to highlight their strengths and weaknesses and also establish the need for the proposed tool. After initial research, there were three websites clearly stood out claiming to be a case study database, these will be discussed in turn.

2.4.1 Embedded EthiCS

Harvard University scholars have been working diligently towards the goal of integrating computer ethics education into technical courses [6]. Embedded EthiCS [4] was established in the autumn of 2016 after students in Barbara Grosz's computer science course on "Intelligent Systems: Design and Ethical Challenges" argued that more computer science courses should include discussions of ethics. With the help of Philosophy Professor Alison Simmons, their objective was not to create separate ethics courses but to integrate ethical considerations into current computer science courses. Embedded EthiCS emphasises a "more integrative approach to incorporating ethical reasoning into computer science education" [6]. The program acknowledges the limitations of independent ethics courses, which fail to connect the social consequences of technical knowledge to which students are currently being exposed. Integrating computer ethics into technical courses in a degree programme addresses this issue. The process involves enlisting advanced Ph.D. students or postdoctoral fellows in philosophy who possess a strong background in ethics to develop ethics modules associated with computer science course content. These individuals design the lessons, create assignments, and plan for assessing them. Research at Harvard has shown mixed results on the effectiveness of the programme [6]. Student engagement is higher when the module blends the technical material with ethical issues that are already important to the students. Moreover, the courses were found to be more effective if they are created with close faculty engagement between CS and philosophy professors. Nonetheless, Harvard did encounter some hurdles when establishing this cross-faculty resource. As discussed previously, the hesitation from each faculty to teach in another subject field was evident. Philosophers believed they did not have the knowledge or experience to teach CS concepts to students and support was provided by CS professors when creating the materials. One must also consider the practicalities of requiring collaboration from Ph.D. students or postdoctoral fellows in philosophy to create these teaching modules. Can the creation of these materials keep up with the endlessly evolving technology if they require several meetings before they are approved to be used for teaching?

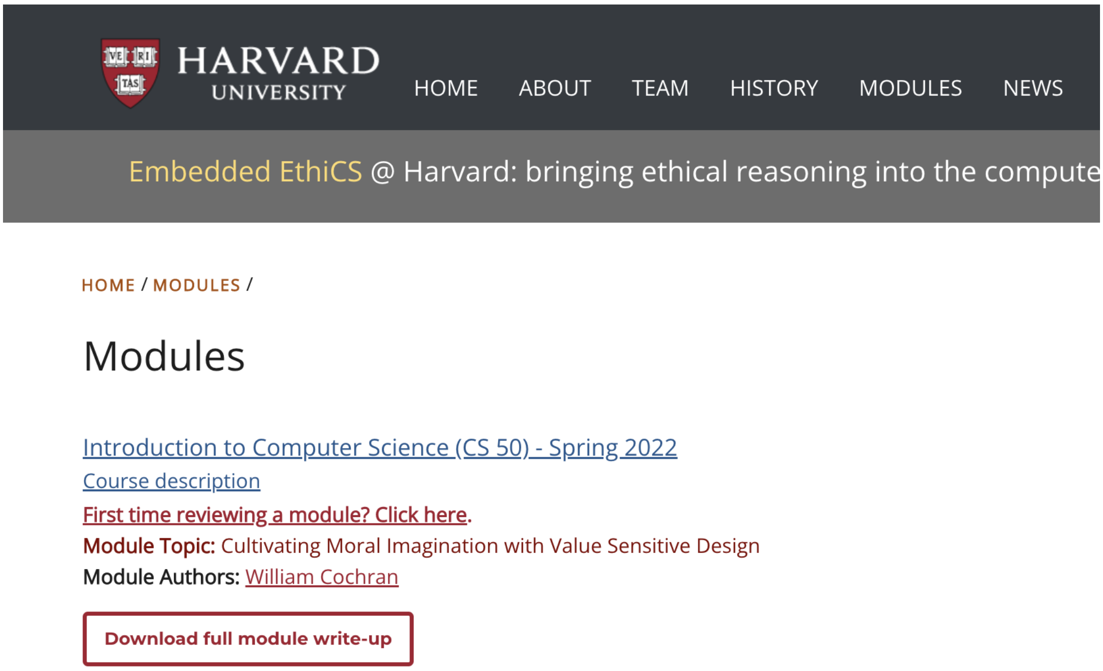
2.4.2 The Embedded EthiCS tool - a closer look

As previously discussed, the Embedded EthiCS tool focuses on a module-based database approach. Lecturers are able to utilise the search bar and filter feature to discover modules exploring ethical issues within the realm of computer science. Upon testing this website, it was discovered that the search bar goes beyond a simple term check and prioritises relevant documents for the user. The assumption is that the search algorithm

[4] https://embeddedethics.seas.harvard.edu/

utilises a text analysis method, such as the term frequency-inverse document frequency (TFIDF) search, which is commonly used in various fields, including information retrieval and machine learning. This advanced search algorithm provides more precise results compared to a standard filter search. Along with a smart search, the website also offers a 'filter by term' option to allow lecturers to only see modules from a specific semester.

As discussed in section 2.4.1, the goal was to integrate ethics modules into ongoing CS courses instead of making a stand-alone course dedicated to the teaching of computer ethics. Thus each entry in the database contains an ethics module, whose title represents the CS course that the ethics module could be integrated into, this can be seen in figure 2.1.


HARVARD
UNIVERSITY
HOME
ABOUT
TEAM
HISTORY
MODULES
NEWS
Embedded EthiCS @ Harvard: bringing ethical reasoning into the compute
HOME / MODULES /

Modules

Introduction to Computer Science (CS 50) - Spring 2022

Course description

First time reviewing a module? Click here.

Module Topic: Cultivating Moral Imagination with Value Sensitive Design

Module Authors: William Cochran

Download full module write-up

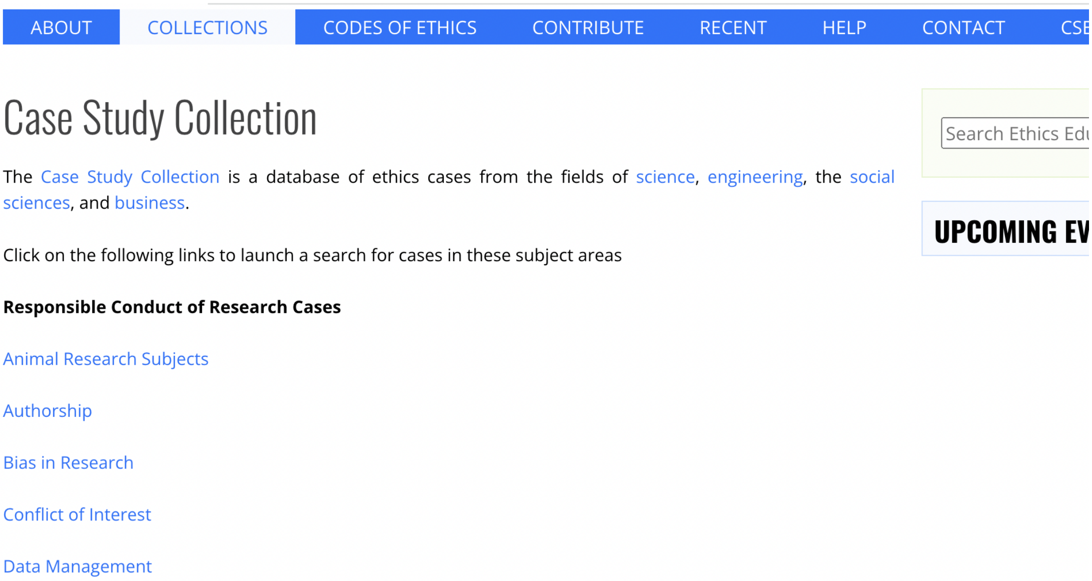
Figure 2.1: Embedded EthiCS; Example list of ethics courses for whom ethics modules have been created.

Each module, within this database, contains a plethora of information with reference to the module such as the course overview, module overview, module goals, key philosophical concepts, and an example of how to implement each ethics module into a course.

Despite its robustness and having undergone extensive testing at Harvard for several years, the following criticisms:

- Long creation time for a module due to the sheer amount of information in each one. Perhaps leading to less up-to-date materials, which according to Quinn [21] is less engaging for pupils.
- Co-creation of materials with the Philosophy department increases the complexity of creating materials.

2.4.3 Ethics Education Library - Illinois Institute of Technology



Case Study Collection

The Case Study Collection is a database of ethics cases from the fields of science, engineering, the social sciences, and business.

Click on the following links to launch a search for cases in these subject areas

Responsible Conduct of Research Cases

Animal Research Subjects

Authorship

Bias in Research

Conflict of Interest

Data Management

Figure 2.2: Ethics Education Library; The main screen with folders of ethics case studies.

Another resource that was found was created by the Illinois Institute of Technology. Established in 1976, the Ethics Educations Library's[5] mission was "to educate students as responsible professionals, to reflect on the wider implications of scientific progress". The resource offers a range of ethics materials for scholars and lecturers who intend to develop "training and instruction programs" for students. The case studies however are nested in folders, increasing the time and effort in finding what one is looking for. Figure 2.2 above shows this database.

2.4.4 AIAAIC repository

The AIAAIC [6] is an "independent, free, open library that identifies and assesses 950+ incidents and controversies driven by and relating to AI, algorithms, and automation" The database consists of a flat spreadsheet of case studies. Each case study is given tags so that a user can sort by tags and find what they are looking for. However, due to the nature of the spreadsheet, it is difficult to search for specific keywords to find specific case studies.

2.4.5 The Ethics CS Portal niche

The previous tools were unique in their own way. Embedded EthiCS was a repository with lots of extra information on how to integrate each module into a course, and the AIAAIC had lots of case studies but no way to search through them.

Ethics CS Portal aims to fix the flaws that appear in some of the other resources. As discussed in the final remarks of section 2.4.1 the Embedded EthiCS resource

[5] http://ethics.iit.edu
[6] https://www.aiaaic.org/home

follows a module-based approach where each entry in the repository is a comprehensive ethics module (usually based on a case study) which details exactly how this can be incorporated into an existing CS course. The fields in each entry might include potential class activities, ethical theories the class should be discussing, and questions to prompt discussion. All of this information increases the creation time for each module and thus might not keep up with the ever-evolving nature of technology.

The Ethics CS Portal will follow a case study approach due to its simplicity and effectiveness as established in section 2.3. This will reduce the creation time for entries in the repository as the information will only focus on the case study itself.

2.5 The importance of teaching CS/AI ethics in higher education

It is not rare to hear about or read in the press about companies and whole industries that have been under scrutiny due to technology that has been implemented without consideration of the social and economic impacts it could have. Software engineers and computer scientists are essential in building a safer, more efficient, and healthier future. *All* developers should be asking themselves not only "can I implement this software" but also "*should* I implement this software" and "is it ethically viable?" and "will there be any negative consequences and if so, how do we mitigate them?".

2.5.1 What happens when ethics is not considered?

As mentioned previously the press are quick to accuse technology of being "ethically wrong". Scandals like Facebook-Cambridge Analytica [1] or the Volkswagen "diesel dupe" [9] are not uncommon to hear about in the news. A recent article [16] provides some insight into the culture at Silicon Valley, which could be the reason behind these long lists of tech scandals. The article considers the notion that many technical experts "position themselves as the actors best suited to address ethical challenges, rather than less technically-inclined stakeholders, including elected officials and advocacy groups." [16]. Although there are procedures to help designers identify potential ethical problems with their implementations, the designer might not have the ability to put themselves in someone else's shoes (someone who might be very different from a software engineer or computer scientist) and discover potential ethical harms that their technology could have. Including stakeholders in discussions will allow the technology to be observed from a different point of view.

2.6 Methods of human-computer interaction (HCI) evaluation

In the following section, common methodologies for gathering requirements and evaluating a design product will be considered.

2.6.1 What is human-computer interaction?

Human–computer interaction (HCI) as the name suggests is the study of how humans interact with computers, in particular focusing on the design of computer technology to make it easy to use for humans. A major part of this field of study is designated to design and the idea that systems should be usable, efficient, effective, and satisfy the needs of the end user. [2]

2.6.2 Requirements gathering techniques

2.6.2.1 Interviews

Interviews are one of two methods of survey research, the other being questionnaires [7, p. 102]. These can be conducted in person or online.

There are two types of interviews structured and unstructured. Structured interviews follow a script whereas unstructured interviews can be adapted at the time. Unstructured interviews have the benefit of seeming more natural and make participants more comfortable sharing their views whereas structured interviews are perhaps more formal but allow for more control over the flow of the interview and ensure the answers are more insightful.

2.6.2.2 Questionnaires

Questionnaires are the second method of survey research [7, p. 140]. Usually written up as a list of short questions to gather insight from different target groups about their perceptions, feelings, or attitudes.

Benefits of questionnaires include:

- Simple to produce and distribute.
- Highly scalable, can be distributed online, lots of data can be gathered.
- Easy to perform data analysis as answers to some questions might be quantitative.

Drawbacks of questionnaires include:

- Highly reliant on question-wording, to obtain the most insightful results as possible.
- Difficult to obtain follow-up information from the participant, if the answer is vague or not helpful it is hard to rectify.

2.6.2.3 Focus groups

Focus groups are a qualitative method used to gather insight into feelings, thoughts, and feedback on a specific product or system.

2.6.3 Evaluation techniques

2.6.3.1 Cognitive walkthrough

The notion of a cognitive walkthrough (CW) was introduced by Clayton Lewis and his colleagues in 1990 [13]. This first CW was intended to be used by evaluators on systems that can be used with little to no training. Lewis and his colleagues decided that the first rendition of the CW was unsatisfactory and therefore two new versions were created, the last of which being CW3 in 1994. CW3 calls upon the evaluator to put themselves in the shoes of a potential end user and imagine a scenario in which the user is trying to accomplish a task. To complete a task the evaluator must follow certain steps toward their goal. At each step towards the main goal, the evaluator must ask themselves four questions about their progress.

- Will the user try and achieve the right outcome?
- Will the user notice that the correct action is available to them?
- Will the user associate the correct action with the outcome they expect to achieve?
- If the correct action is performed, will the user see that progress is being made toward their intended outcome?

The answers to these questions are then analysed and the usability of the website can be determined.

2.6.3.2 Heuristic evaluation

Heuristic evaluation, as defined by Neilson, is a “usability engineering method for finding the usability problems in a user interface design” [18]. This evaluation process involves having a small set of evaluators (optimally between 3 and 5) follow a set of heuristics or principles of acceptable design. Neilsen’s 10 heuristics are listed below:

- Visibility of system status.
- Match between system and the real world.
- User control and freedom.
- Consistency and standards.
- Error prevention.
- Recognition rather than recall.
- Flexibility and efficiency of use.
- Aesthetic and minimalist design.
- Help users recognise, diagnose, and recover from errors.
- Help and documentation.

This method provides a general way to re-generate a design according to the issues that violate the usability heuristics, however it does not provide a detailed diagnosis of why and how to amend a design issue like a cognitive walkthrough does.

Depending on the type of system that is being tested participants have the option to carry out the evaluation in a guided or non-guided manner. If the system's intended target audience is the general public or has been designed as a walk-up-and-use interface then the evaluators can carry out the evaluation alone by filling out a report. However, if the system is more complex and the evaluators are inexperienced in the domain then they will need to be guided through the process.

2.6.3.3 Heuristic walkthrough

A heuristic walkthrough is an evaluation method that incorporates the previous two ideas into one. It brings the benefits of a scenario-based cognitive walkthrough, whilst also including the heuristic-based evaluation. Evaluation using this process is carried out in two phases, firstly by providing the evaluator with a set of tasks to carry out, whilst in phase two the evaluation is less strictly bounded and the user is free to explore all of the system, which hopefully, with the knowledge from the previous stage is thus made easier. During the second stage, the user notes down any part of the system that violates Neilsen's 10 usability heuristics. Sears [23] found that a "heuristic walkthrough can find more usability problems than a cognitive walkthrough v3 while producing fewer false usability problems than the heuristic evaluation method".

2.6.3.4 User acceptance testing

User acceptance testing (UAT's) is conducted during the final stage of software development testing, if the software undergoes a successful UAT and amendments are made by the developers, the software can be released to the public.

Firstly, a set of acceptance criteria is laid out by the developer which must be met by the software to be considered ready for deployment. Subsequently, a set of test cases will be provided to the end user to test whether the software meets the acceptance criteria. The results can then be analysed and any test cases that failed can be repaired.

Chapter 3

Methodology

3.1 Choosing the software development method

Agile software development represents the idea of nimbleness and flexibility in development. This technique was chosen for this project as it enables requirements to be altered throughout design and implementation. The Agile Software Development Manifesto [1] was published in 2001 by a group of software practitioners in 2001. The main points that the manifesto emphasised were:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

The final point was followed in this project as there were many occasions during implementation where new ideas were formed and requirements were added and removed due to timing issues, researching requirements from similar websites in the field, and discussions with the supervisor of this project.

3.2 Choosing the development tools

3.2.1 Designing the prototypes

After initially sketching designs on paper it was then necessary to move to a more sophisticated form of mock-up, namely a prototype that could be used to gather feedback on the design and usability of the website.

Proto.io [2] is a prototyping software used in this project due to its large component and icon library, ready-made templates, and its ability to create interactive live screen transitions with a simple drag and drop.

[1] agilemanifesto.org

[2] https://proto.io/

3.2.2 The programming language

The decision was made to use React as the development environment for the website. React is a Javascript library developed by Meta and is the basis of many large applications like Netflix, Paypal, and Airbnb. React shines in its simplicity to set up, its vast choice of public libraries for integrating features into an application with significantly less code than vanilla Javascript, and its component structure. React improves code structure and hence readability as it uses a component-based architecture. React comes built-in with one main App component (which is called the Root Component) but developers can nest components inside of each other reducing the need for files with hundreds of lines of code. Moreover, debugging is also made significantly simpler by having a component-based architecture. If a bug appears in the code, React indicates exactly which file is causing the error and developers can quickly rectify the issue. Moreover, Reacts' new hook API also works to make code more readable, concise, and clear. A hook in React is essentially a new way to write components without having to create a class with state data. The useState() hook was one that was used plentifully during the creation of the tool. This allows a constant to be used and updated without having to write a class and setter method, it is all done by the hook library. If you need a constant in a different component, passing props down through other components is no longer necessary, developers can create their own custom hooks to fetch values from a database and use them throughout the application.

3.2.3 The database

The database used in this tool is called Firebase. "Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world."[3] Firebase offers an all-in-one solution for web development including features such as authentication, database, and hosting.

3.3 Requirements gathering

3.3.1 The questionnaire

Another method of requirements gathering that was conducted as part of this project was aimed at collecting requirements from the end-users themselves. This was achieved by creating a Microsoft Form questionnaire and analysing the answers. The aim of the questionnaire was to gather requirements of potential features that could be incorporated into the website. In total seven participants were involved in this stage of the study. Upon completion of the data analysis, the requirements were documented and prioritised accordingly (see section 4.2.2).

[3] https://firebase.google.com/

3.3.1.1 Deciding upon the end users

As stated in the website description, the tool initially was intended to be for lecturers who teach a computer ethics module at the University of Edinburgh or are generally interested in tech ethics. The link to the Microsoft Form would be sent out via email and so firstly, a list of participants would need to be collated from the university website. Once this was achieved, the email was sent to the selected participants across all departments of the university, and responses were gathered. The list of potential participants was made up of people from different departments at the university, 47% from the law school, 23% from informatics, 11% from the business school, 8% from a course titled ‘Data Ethics, AI and Responsible Innovation’. The others had varied interests and backgrounds, one participant was a PhD college of art student, and another lecturer majored in social and political science. The diverse background of departments involved with AI/tech ethics highlights its relevance and importance in the modern world.

3.3.1.2 Designing the questionnaire

Creating clear, insightful, thought-provoking questions would be crucial to obtaining the most detailed answers which would in turn lead to a greater understanding of the user requirements and the development of a useful application for them at the end. The central aim of the questionnaire was to gather insight into the participants’ current methodology of obtaining computer ethics resources (prior to the implementation of the Ethics CS Portal). Subsequently, it was important to understand where the participants’ current methodology is effective and where it falls short. This was intended to produce an understanding of the user’s needs. Subsequent questions were aimed at the teaching procedure in computer ethics pedagogy. By asking the participants how they usually teach computer ethics, these results can be compared to the pedagogical methods previously discussed in section 2.3. Moreover, it was also worth also asking the participants whether they write their own descriptions/design their own materials based on a case study or whether they like a more module-based approach like the one seen in the Embedded EthiCS tool. This would back up my hypothesis that for those participants who have their own teaching styles and usually prefer to create their own materials, the Embedded EthiCS cross-faculty approach might not be the most effective way to integrate computer ethics. Three questions in appendix A were straightforward yes/no answers in order to gain insight into the usefulness of the features that were gathered from exploring other case study databases in section 2.4. Finally, the last question was intentionally left as very open-ended one to allow participants to relay any final thoughts/features they would find useful in the application.

3.3.1.3 Data analysis

After the data collection phase, the data was analysed. Due to the qualitative nature of the data, some of the answers to the questions were lengthy and needed to be summarised in a few words so that similar responses could be grouped together to find requirements of greater importance. Thus a mixture of quantitative and qualitative analysis was performed to construct a list of requirements, which would then be implemented in the tool.

3.4 Human computer interaction design evaluation

Once the requirements had been gathered from the previous section, work on design commenced.

An iterative approach to design was carried out throughout this project. The iterative approach allows designers to get ideas onto paper quickly and places user interaction at the heart of the design stage. Done properly in correlation with an agile software development environment, requirements can be easily added or taken away at different stages in the design process with ease.

Firstly, low-fidelity prototypes were used in the form of rough paper sketches to set the groundwork for the application. This meant visualising how requirements gathered from the previous research would look on the website. The next stage in the iterative process was to create an interactive design using medium-fidelity prototypes using software. This was a natural step forward in the design process as it meant that ideas could be more easily manipulated than having to re-draw entire screens by hand.

The software used to create these interactive prototypes was Proto.io. Many other tools such as Figma, Sketch and Azure, exist however, due to the time constraints of the project these were not chosen as it would have increased production time by a noticeable factor due to lack of experience with these pieces of software.

The original plan was to perform a formative design evaluation on the Proto.io prototype with former Human-Computer Interaction (HCI) students at the University of Edinburgh. However, this plan was amended because it was foreseen that the prototype would be overly complicated to implement in React. Due to the time limitations of the project, styling the React application to the exact style of the Proto.io prototype would be too time-consuming and therefore an amended plan for the evaluation of the design was used. The revised plan included coding the website in React (trying to mirror the Proto.io design where possible) before doing the evaluation with the HCI students. This would ensure that the feedback received from the design evaluation related to a design similar to the real website design.

Finally, the feedback from the evaluation was then analysed, and a fourth iteration of the design, similar to that of the final website was developed in React.

3.4.1 First iteration - paper sketches

The first stage in the design process was to establish the foundation of the website through paper sketches. This allowed the requirements to be transferred to a design quickly and efficiently.

Initially, the main features that needed to be transferred across to the mockup sketch were:

- The ability to search for case studies.

- The ability to favourite/'save for later' case studies.
- The ability to obtain more information about a case study like press links, ethical theories involved, and a description of the case study.

Figure 3.1 shows the initial mockup for the home screen. From this figure, it can be seen that the 'Search' feature has been designed by implementing a search bar at the top of the page. Moreover, on each case study, there is a star indicating the ability to 'favourite' a case study. Finally, figure 3.3 is the mockup of the case study screen, where a user will be able to get more information on a particular case study.

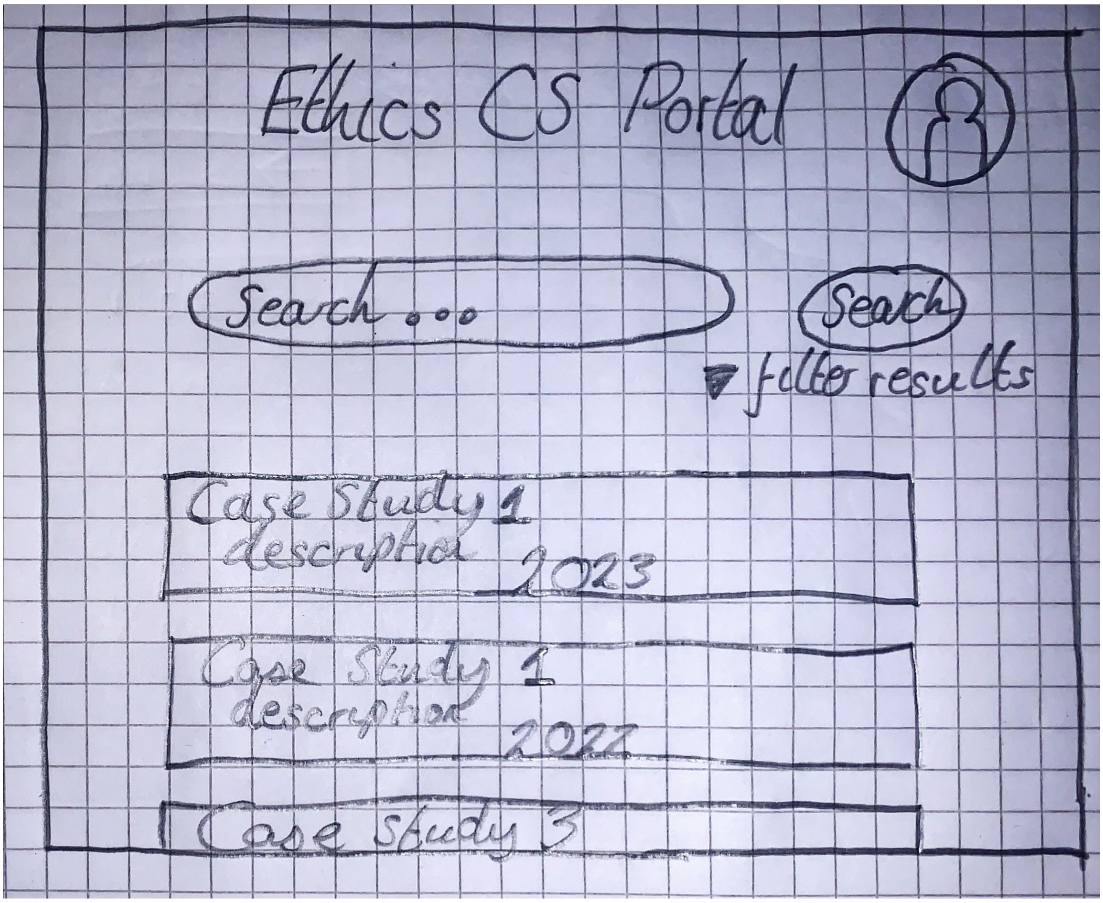


Figure 3.1: Hand-drawn sketch of the dashboard screen of the Ethics CS Portal tool

3.4.2 Second iteration - interactive prototype

Once the sketches had been drawn, the obvious next stage was to create an interactive version of the design, online using software. Proto.io [4] was chosen as the software to create the prototypes due to its vast component library and simplicity in creating live prototypes thanks to its drag-and-drop button interaction method. This prototype, seen in figure 3.4 had the same requirements as previously discussed. Proto.io pre-made components were used for the search bar, filter, and various buttons.

[4] https://proto.io/

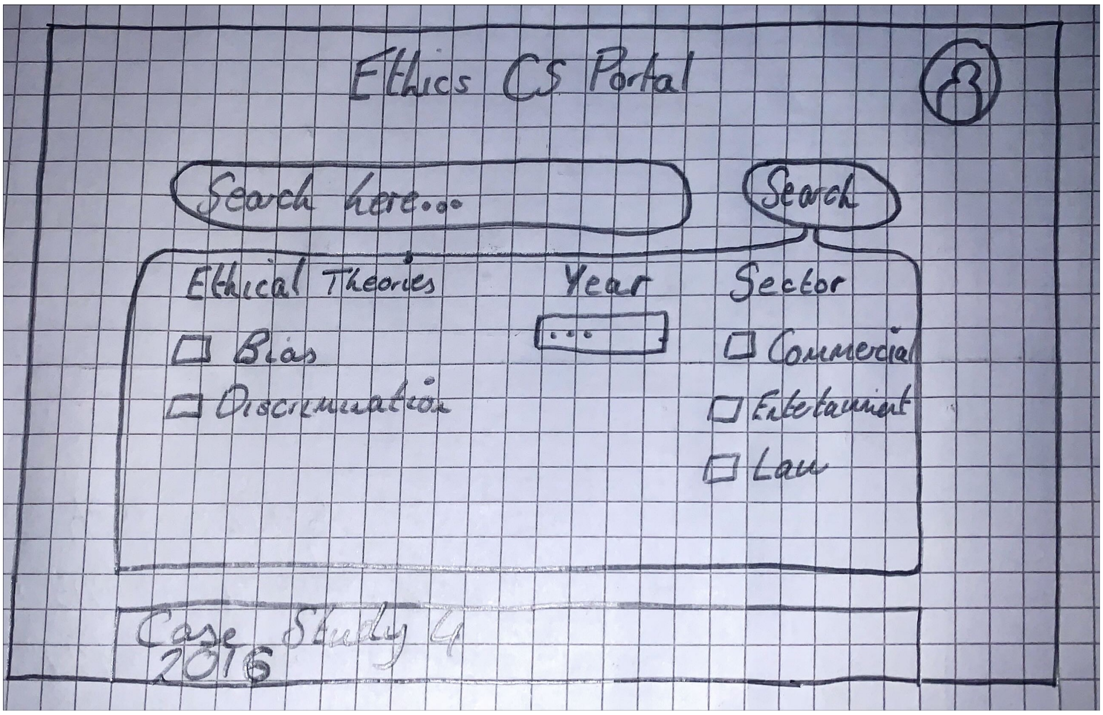


Figure 3.2: Hand-drawn sketch of the filter functionality of the Ethics CS Portal tool

3.4.2.1 Filter section

Initially, a filter section was added to the search area so that a user would be able to filter their search results by things like year, ethical theories, sector, company, and more. The user would check the tags that they would like to filter their search by and these would be taken into account when searching. Eventually, this feature was removed in the next iteration of the design as it was found that the same result could be achieved with a single search bar by using a smart search (TFIDF) algorithm.

After some formative feedback from my supervisor a third iteration of the design was implemented in React so that a live version of the website could be used in the design evaluation.

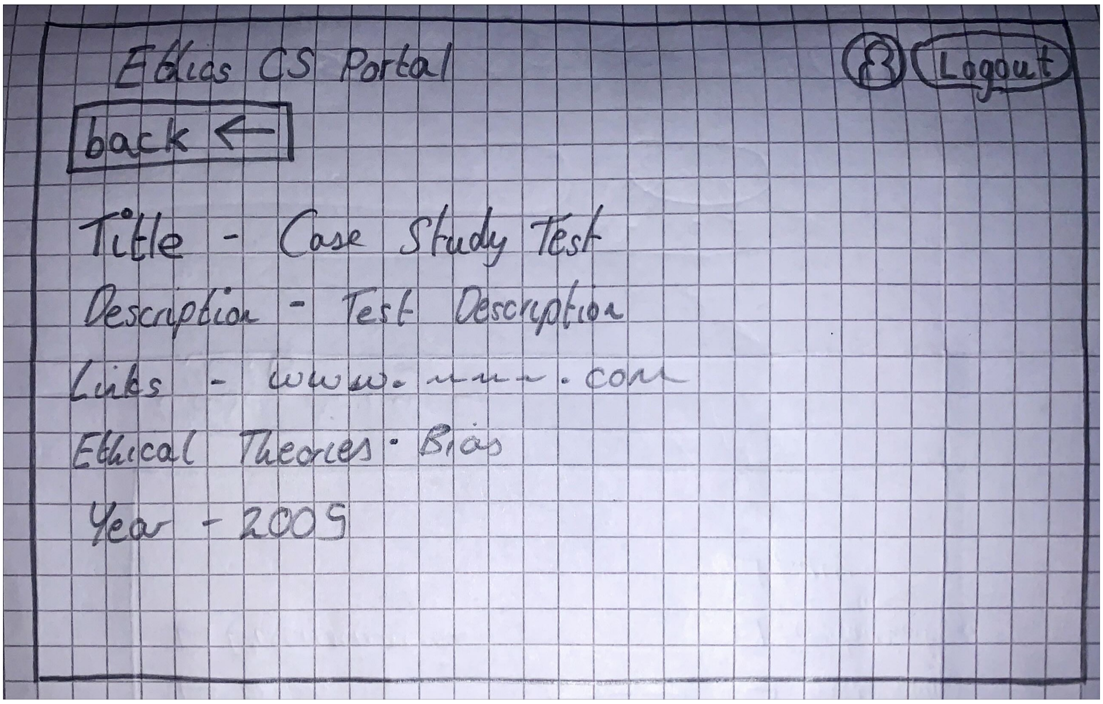
3.4.3 Third and fourth iteration - coded website

These iterations were designed by taking into account the results of the requirements gathering questionnaire and the HCI design evaluation and therefore will be discussed in more detail in section 4.2.2.5 and 4.3.3

As discussed previously the ability to mirror a prototype exactly into code requires more CSS styling in React and therefore it was decided to implement the website trying to resemble the prototype as closely as possible whilst using MUI's [5] and ChakraUI's [6] component libraries.

[5] https://mui.com/

[6] https://chakra-ui.com/



Title - Case Study Test

Description - Test Description

Links - www. - - - .com

Ethical Theories - Bias

Year - 2005

Figure 3.3: Hand-drawn sketch of the case study page of the Ethics CS Portal tool

3.4.4 Heuristic walkthrough with HCI students

Following the implementation of the third iteration in React it was necessary to perform evaluation methods in HCI design to gain feedback at this stage.

3.4.4.1 Aims

The purpose of the evaluation was to gather expert feedback on the usability of the application. Whether the website was intuitive to use and tasks could be followed easily and also whether the application followed Neilsen's 10 usability heuristics [17].

3.4.4.2 Evaluation method

The evaluation method used was a heuristic walkthrough as discussed in section 2.6.3.3. This was chosen due to its efficiency in finding more usability problems and fewer false usability problems than the heuristic evaluation method.

3.4.4.3 Choosing the participants

According to Neilsen's [18] research, the optimal number of evaluators for a heuristic walkthrough is typically between three and five. This is because adding more evaluators may lead to redundant findings and overlapping feedback, rather than uncovering new usability issues. To ensure an effective evaluation process in our project, a heuristic walkthrough was carried out with a team of three evaluators.

The participation criteria for this type of evaluation is flexible and can include a range

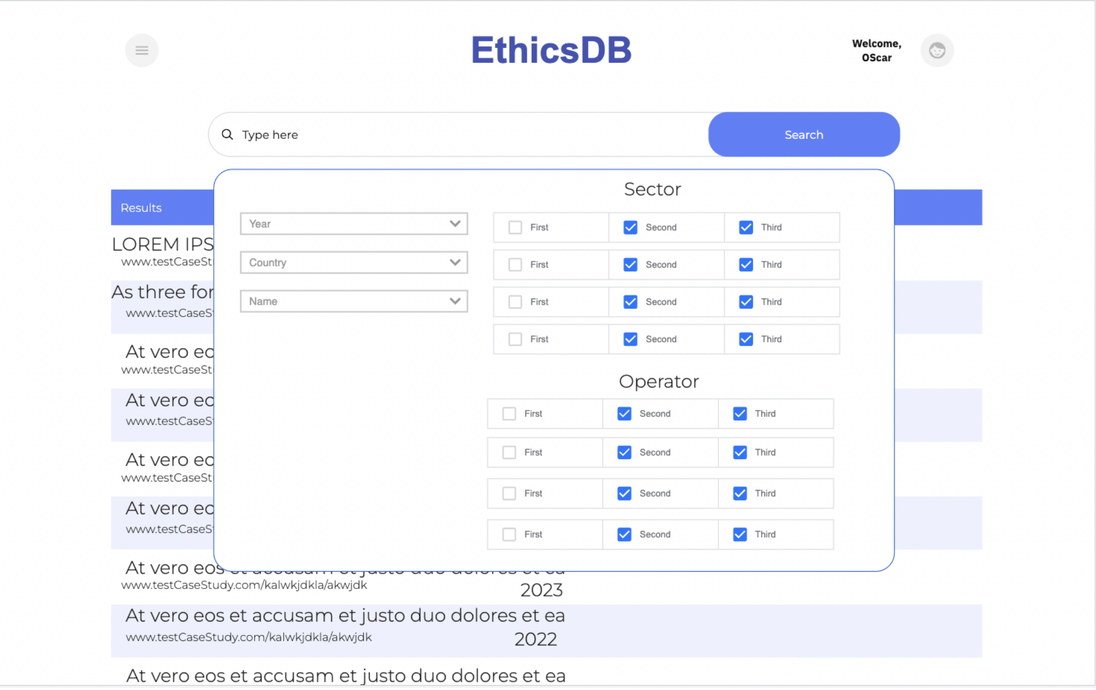


Figure 3.4: Mockup of filter section created with Proto.io

of stakeholders, end-users, the general public, and HCI experts. However, for this particular evaluation, former HCI students from the University of Edinburgh were selected, as they were likely to provide insightful feedback on the design. The main objective of this evaluation was to gather feedback on the website's design and usability.

It was decided not to involve end-users in this stage, as their feedback may have been more focused on whether the website met their specific requirements, rather than providing feedback from a design perspective.

3.4.4.4 Materials for participants

As the intended application of the system is specifically within the field of education, the system might not be intuitive to the HCI students chosen to participate in this study. Hence it was decided that the evaluation process would be guided during an online Microsoft Teams meeting.

An evaluation template, which can be found in appendix B, was composed prior to the evaluation session, allowing for the evaluator's comments to be easily textualised into a format that is readable and facilitates data analysis. This template included a set of 6 tasks intended to test the usability of the system as discussed in section 2.6.3.1, along with a template for the heuristic evaluation section.

3.4.4.5 The process

Once the heuristic walkthrough template had been created, three participants with experience in HCI separately took part in a 30-minute Microsoft Teams call.

The evaluation followed the heuristic walkthrough protocol of first running through the set of tasks that would familiarise them with the system. At this stage, any issues in completing the tasks were noted and assigned a severity rating on the following scale.

- 0 – Not a problem – Evaluator thinks that this is not a usability problem at all.
- 1 – Cosmetic problem only – Need not be fixed unless extra time is available on the project.
- 2 – Minor usability problem – Fixing this should be given low priority.
- 3 – Major usability problem – Important to fix, so should be given high priority.
- 4 – Usability catastrophe – Imperative to fix this before the product can be released.

Afterward, the evaluation proceeded to its second phase, which involved conducting a heuristic evaluation. During this stage, the evaluators were allotted 10 minutes to thoroughly explore all aspects of the system and express any issues they encountered against Neilsen's 10 usability heuristics 2.6.3.2.

These results were recorded in note format and subsequently compiled into a comprehensive and well-documented report.

Once this process had been completed with each of the three evaluators the results were collated so that they could be analysed and issues could be given priority (to be fixed) based on their severity rating.

3.4.4.6 Data analysis

Duplicate issues were removed from the collated spreadsheet and the issues were ordered by severity rating. These issues are discussed in the results section 4.3

3.5 User acceptance testing

In a research paper examining the essential qualities of a successful system, Dromey [3] affirms that a system's readiness is determined by its "correctness, structuredness, modularity, and descriptive properties". However, other testing methods, such as unit and performance testing, focus primarily on the developer rather than the user, resulting in a lack of attention to critical issues like operational procedures and the operating environment [12]. As the end-user has the final say on the quality of the product, testing the product on the final user must be integrated into the software development lifecycle. This integration can be achieved through User Acceptance Testing (UAT).

3.5.1 Aims

The primary goal of the User Acceptance Testing (UAT) was to collect feedback regarding whether the tool fulfilled the requirements outlined in section 3.3. Essentially, this stage determined whether the tool was prepared for deployment. If the end users were satisfied and comfortable that the tool met their needs, it would be considered ready for deployment. However, if the tool fell short, more effort would be required to fix the usability or feature issues before it could be deployed.

3.5.2 The process

3.5.2.1 Creating the UAT documents

Firstly, a template of the UAT document was drafted. This document needed to draw up the success criteria for the tool. Once the success criteria had been established it was necessary to identify some test cases that would assess said criteria. The UAT document can be seen in appendix C.

3.5.2.2 Performing the UAT

Initially, an email was sent out to the list of participants which was gathered in section 3.3.1.1. This email contained two options regarding how the participants could conduct the evaluation. The first option was a guided UAT where the participant would join a Microsoft Teams call and be subsequently asked to execute the test cases, establish whether they passed or failed, and give any further general comments. Secondly, the option of completing the UAT alone was also proposed, if this method was preferred the participant could test the tool at their own pace and answer the questions on the UAT template and share their feedback. Two of the participants decided to join a Microsoft Teams call and one of them preferred to complete the document in their own time.

3.5.2.3 Analysing the results

The results from the UAT test cases were collated and the cases were ordered from highest to lowest fail rate. This would ensure that the issues with the highest fail rate would be given a higher priority to be fixed. A spreadsheet was used to analyse the data as the information could be displayed in a well-organised and easy-to-read table.

Chapter 4

Results

4.1 Analysing the implementation environment

4.1.1 The programming language

The React programming environment integrated almost seamlessly with Firebase. A few problems arose when executing Firebase commands that were deprecated and had been recently replaced by new methods of authentication and database read/write commands. Due to the recent nature of the changes from Firebase, solutions to Firebase errors found online sometimes refer to older versions of the system.

4.2 Requirements gathering

4.2.1 Researching other similar systems

As discussed in section 2.4, it was important to investigate other tools to gather the types of features expected in a computer ethics repository and also identify the common pitfalls of existing tools so that the Ethics CS Portal could be designed to rectify them.

4.2.1.1 Key insights

This method of requirements gathering was helpful as it allowed the drafting of some requirements at an early stage of design. These requirements were:

- Smart search bar for accurate results.
- Case studies ordered by date, most recent first.
- Tags for categorical search.

4.2.1.2 Implementing the term frequency inverse document frequency (TFIDF) smart search

This algorithm was implemented into the search bar of the Ethics CS Portal to give more accurate results to the user. The algorithm calculates values for each word in a case study by a process of computing the frequency of a word in a particular case study divided by the percentage of case studies that the word appears in. For example, the word ‘the’ appears in most documents and therefore is not a strong indicator of what a case study actually discusses, therefore it would be given a low TFIDF score. A word like ‘Amazon’ however probably does not appear in many documents and therefore if it did appear in one document it would be given a high TFIDF score as it would be a big indicator that the particular case study discusses the company Amazon in some way. This is the basis of the TFIDF algorithm and works more effectively than a standard search, whereby results are shown to the user based on the entire query. The TFIDF algorithm works on words or ‘tokens’ and so, if a user searches for ‘algorithmic bias in job recruitment’ it will return case studies to the user based on the words in the query rather than the entire query itself. Words with high TFIDF scores implicit a strong relationship with the text that they appear in “suggesting that if that word were to appear in a query, the document could be of interest to the user.” [22].

4.2.2 Questionnaire

The following section discusses the main results from the requirements gathering questionnaire discussed in section 3.3.1.

4.2.2.1 Key insights

After performing data analysis on the answers from the questionnaire, key requirements were extracted, as below:

- Easy-to-add case studies, ensuring minimal excess information.
- A way to store/bookmark case studies making it easier to re-find them.
- Community-driven database.

The following requirements were also gathered from this questionnaire, however, these were not implemented in the final tool due to timing constraints.

- Popularity score.
- Tags for which courses have used a case study already.

Priority was given to features that were essential for the tool to achieve the main aim of this report, the above requirements could and should be added in future work in this field and this will be discussed in section 5.2.1 and 5.2.2.

Each requirement that was implemented in the final tool will now be discussed in more detail.

4.2.2.2 Easy-to-add case studies, ensuring minimal excess information

The majority of participants use quick and effortless methods to find case studies, such as browsing the Internet and social media, or receiving news articles from colleagues. They believe this approach is effective and ensures that they obtain the most recent and relevant examples to be used as case studies. In addition, a significant number of participants (87%) create their own case study descriptions and develop their own teaching materials (71%). As a result, using the Embedded EthiCS module-based approach, which provides information such as example class activities, course overview, and module goals that may not be relevant, was deemed unnecessary. To ensure that the website's materials remain up-to-date with the constantly changing technology landscape, the process of adding a case study to the portal must be as quick and easy as possible.

4.2.2.3 A way to favourite/bookmark case studies making it easier to find them again

Participants also communicated that one of the main troubles with searching the internet/social media is that it can be time-consuming to re-find a case study again. “I forget stuff, or spend ages trying to find it again!”

Participants shared some of their methods of saving case studies, one participant noted “I might save the link in a Word document” and another wrote “I keep screenshots and URLs of articles in a LaTeX document”. Therefore, this requirement was drafted to mitigate this issue.

4.2.2.4 Community-driven database

To keep the website materials as up-to-date as possible it will also be necessary to make it a community-driven website where any lecturer can add material that they come across. The website will act as a hub where lecturers can share their materials and access material that other staff have come across making it more of a shared repository similar to the one seen in the AIAAIC 2.4.4.

4.2.2.5 Third iteration - website

With these requirements in mind, a third iteration of the website was designed and coded using React.

In order to make saving case studies more convenient for lecturers, a star icon has been incorporated, see figures 4.1 and 4.2, which allows them to easily add items to their favourites with a simple click. Additionally, figure 4.3 highlights the required fields for adding a new case study. It's worth noting that the process has been designed to be quick and easy, with only a few essential fields to complete.

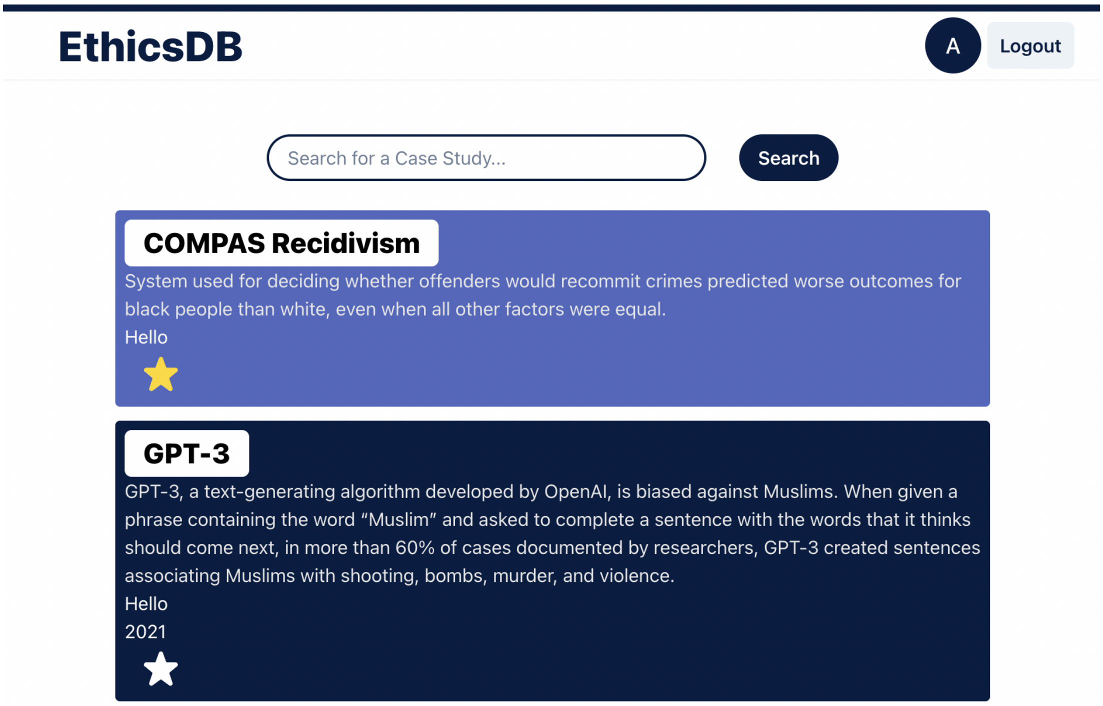
4.3 Human computer interaction design evaluation

4.3.1 Key insights

Following the HCI heuristic walkthrough with three participants, the data was analysed and collated, the issues regarding the design were grouped into Neilsen's heuristics and are shown below

4.3.1.1 Aesthetic and minimalist design

- Logout and profile are potentially too close together.
- Margin for text in the dashboard is too small.
- Hovering should turn star gold.
- Case study page information placement makes it hard to read.



COMPAS Recidivism

System used for deciding whether offenders would recommit crimes predicted worse outcomes for black people than white, even when all other factors were equal.

Hello

★

GPT-3

GPT-3, a text-generating algorithm developed by OpenAI, is biased against Muslims. When given a phrase containing the word "Muslim" and asked to complete a sentence with the words that it thinks should come next, in more than 60% of cases documented by researchers, GPT-3 created sentences associating Muslims with shooting, bombs, murder, and violence.

Hello

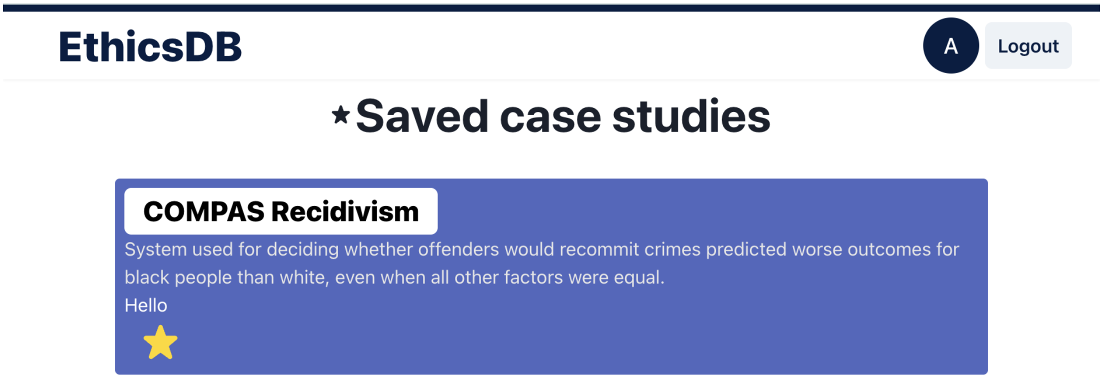
2021

☆

Figure 4.1: Ethics CS Portal dashboard page

4.3.1.2 Consistency and standards

- Favourite Star, only being available on the home page made it hard for the user to favourite a case study after they viewed it as they would need to go back and search for the same case study again to favourite it on the dashboard.
- URL's are not highlighted and underlined.
- Inconsistencies in capitalisation.



EthicsDB

★Saved case studies

COMPAS Recidivism

System used for deciding whether offenders would recommit crimes predicted worse outcomes for black people than white, even when all other factors were equal.

Hello

⭐

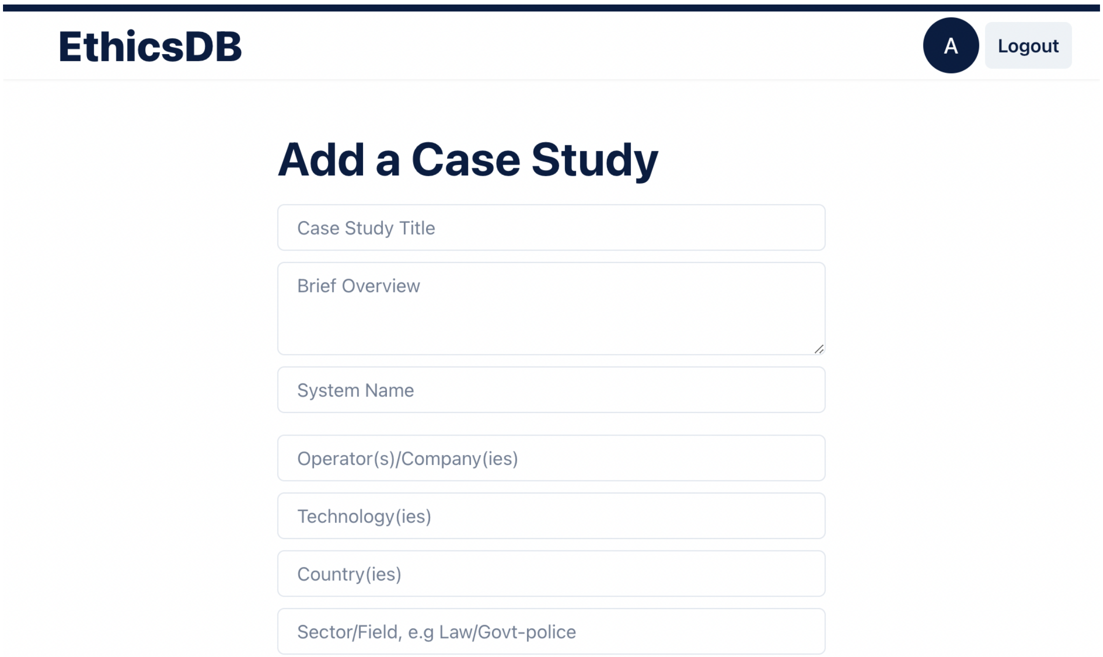
Figure 4.2: Ethics CS Portal favourites page

4.3.1.3 Flexibility and efficiency of use

- Navigation using main 'Ethics CS Portal' text (in the top left) is not intuitive.
- Pressing the URL link re-directs the user to an external site, instead of opening the page in a new tab.
- Add a case study button is too nested, meaning that to complete a simple task users complete unnecessary clicks.

4.3.1.4 Help and documentation

- No clear documentation to clarify what type of information the input fields on the 'add a case study' page should contain.



EthicsDB

Add a Case Study

Case Study Title

Brief Overview

System Name

Operator(s)/Company(ies)

Technology(ies)

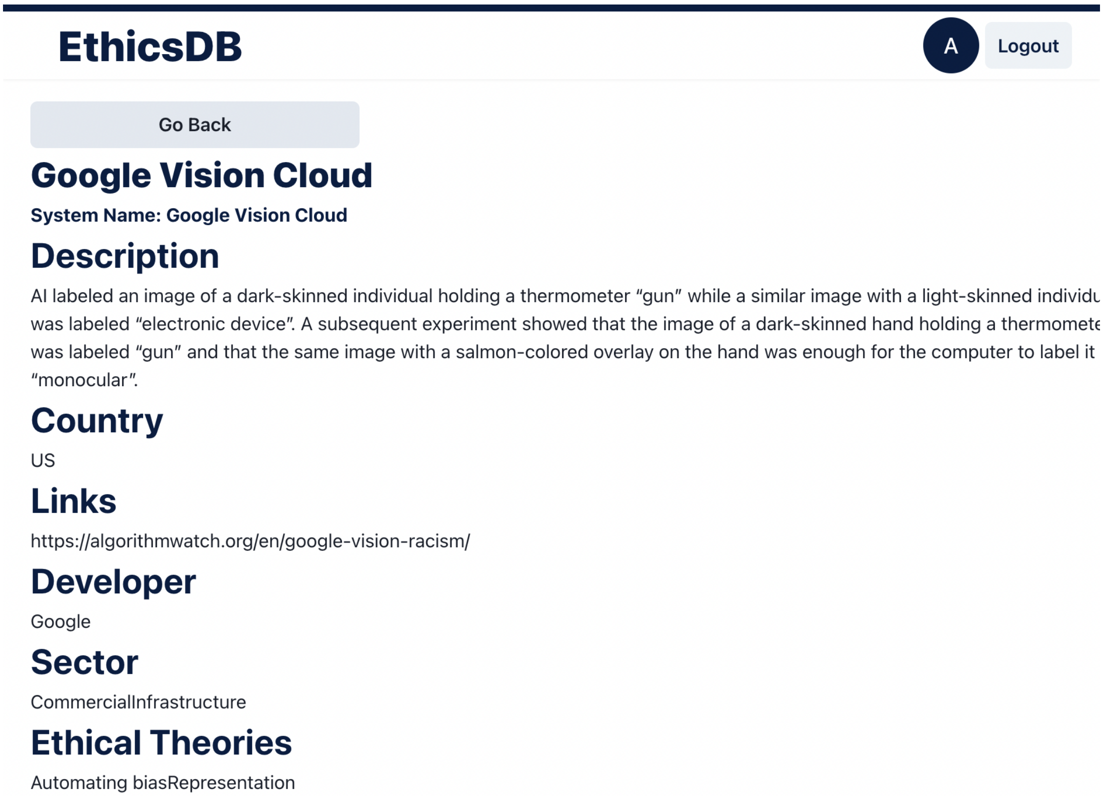
Country(ies)

Sector/Field, e.g Law/Govt-police

Figure 4.3: Ethics CS Portal add a case study page

4.3.1.5 Match between system and real world

- Naming of Press Links perhaps not intuitive.
- No arrow on the go back button (from case study page).
- No indication that favourites are found on the profile.
- Doesn't search for case study when a user presses enter after typing a query in the search bar.
- Star for favourite was not intuitive, perhaps a bookmark symbol would be better, and the favourite section should be renamed 'saved case studies'.



Google Vision Cloud

System Name: Google Vision Cloud

Description

AI labeled an image of a dark-skinned individual holding a thermometer "gun" while a similar image with a light-skinned individu was labeled "electronic device". A subsequent experiment showed that the image of a dark-skinned hand holding a thermomete was labeled "gun" and that the same image with a salmon-colored overlay on the hand was enough for the computer to label it "monocular".

Country

US

Links

https://algorithmwatch.org/en/google-vision-racism/

Developer

Google

Sector

CommercialInfrastructure

Ethical Theories

Automating biasRepresentation

Figure 4.4: Ethics CS Portal case study single view page

4.3.1.6 Recognition rather than recall

- Favourites only being on the profile page - was not intuitive to access.

4.3.1.7 User control and freedom

- Only the white button on the case study is clickable, this is not intuitive for the user as they would expect that clicking anywhere on the case study box would navigate them to the case study page.

4.3.1.8 Visibility of system status

- Search is not a live search, a user is not returned with results as they type. Results are only returned once a user presses the search button.
- Visibility of system status is sometimes hard to decipher, especially when notifications are not provided to the user after completing a task, namely, adding a case study.
- No prompt that the user must press enter to add multiple values in the 'add a case study' fields that may contain more than one value.

4.3.2 Discussion of the findings

The results gathered from former HCI students were insightful and covered many different aspects of design, not just the more easily identifiable aesthetic issues. The students found issues that spanned 8 out of the 10 heuristics of design.

Some interesting issues were brought to the foreground. The 'favourite/save for later' requirement gathered in section 4.2.2.3 had some flaws, for example, a user was only able to 'favourite' a case study on the dashboard page and not the single view case study page, making it more difficult to adhere to the requirement of saving a case study for later.

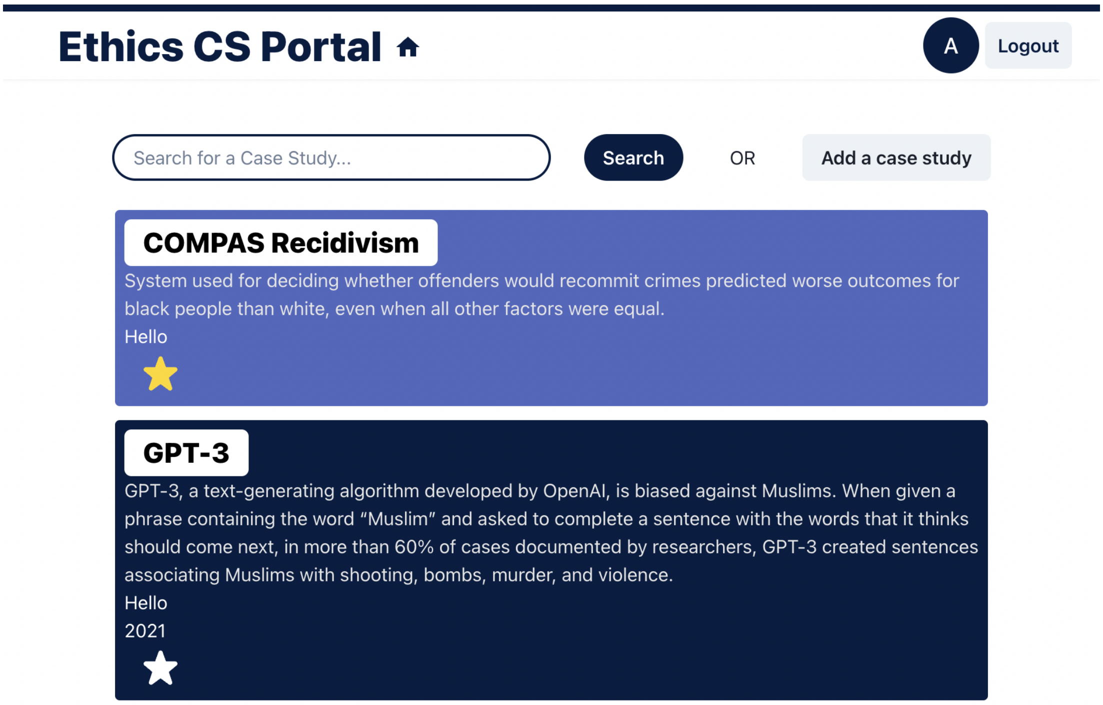
Systems that adhere to the match between the system and real-world heuristics should remember that the way people perceive technology is shaped by their past interactions with digital devices and their real-world experiences. Design problems that violated this heuristic included: naming conventions; not adding an arrow to the back button located in figure 4.4; the search bar not searching when the user presses enter; and the star icon not conveying the idea of a 'bookmark/save for later' feature, as seen on other major applications.

Finally, issues that didn't comply with the visibility of system status heuristic were found when a participant was adding a case study to the database as seen in 4.3. When a user clicks to add a case study there was no notification that the user's request had been fulfilled. Moreover, when inputting the fields like operator(s), technology(ies), and country(ies) it was not clear that the user was required to press enter to add their value to the field.

4.3.3 Fixing the issues

These issues were amended before the next evaluation stage. An example of a redesigned screen can be seen in figure 4.5 where the heuristic violation of the 'add a case study' button only being visible on the profile tab is amended. A user also now has the option of adding a case study from the dashboard page.

This evaluation provided an intermediary step between the initial design and the user acceptance test which will be discussed in the next section.



COMPAS Recidivism

System used for deciding whether offenders would recommit crimes predicted worse outcomes for black people than white, even when all other factors were equal.

Hello

★

GPT-3

GPT-3, a text-generating algorithm developed by OpenAI, is biased against Muslims. When given a phrase containing the word "Muslim" and asked to complete a sentence with the words that it thinks should come next, in more than 60% of cases documented by researchers, GPT-3 created sentences associating Muslims with shooting, bombs, murder, and violence.

Hello

2021

☆

Figure 4.5: Example of fixing the Recognition Rather than Recall

4.4 User acceptance testing

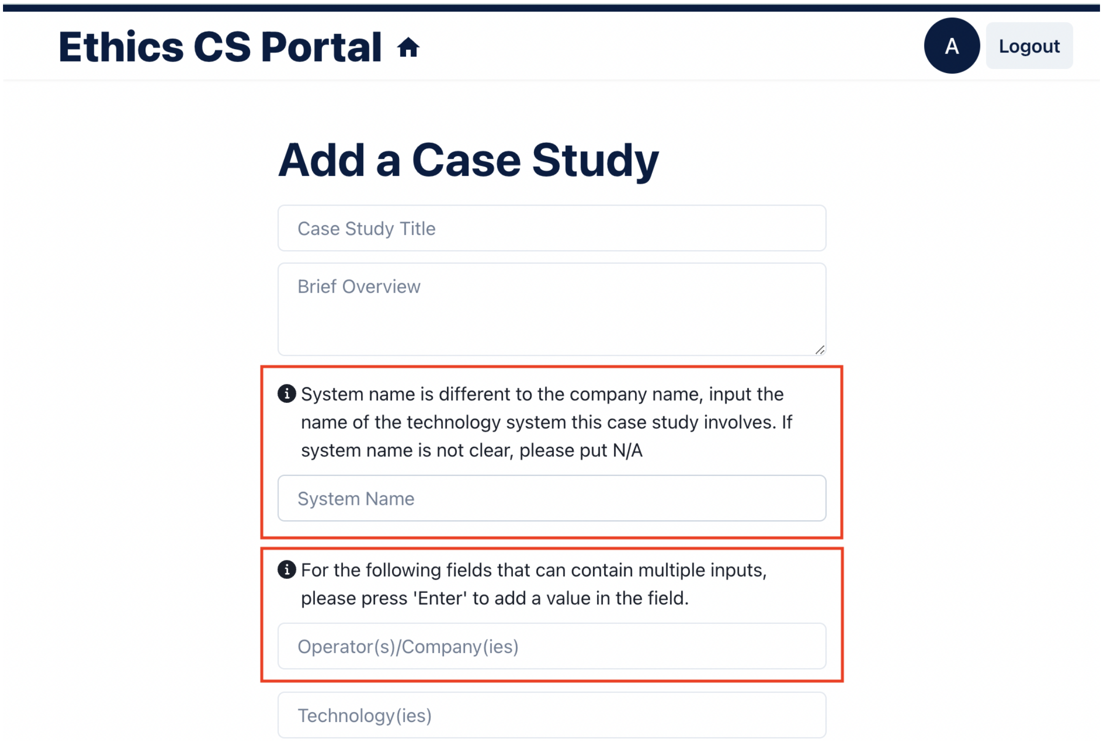
4.4.1 Key insights

The following results were gathered from the UAT carried out with three participants. The outcomes from these results would determine whether the application was ready for deployment or if there were any bugs, features, or usability issues that needed to be amended prior to release.

4.4.1.1 Add a case study issue

The most prominent issue, with a pass rate of 33%, was concerning the ability to add a case study to the database. Participants failed to recognise that these fields could handle multiple values and that these values could be added by pressing the 'Enter' button. Figure 4.6 highlights this problem. As seen in the figure, the terms US USA have not been added as a value as the participant had not pressed the 'Enter' button. This issue produced case studies being added to the database with missing information. Furthermore, participants thought that separating multiple values by commas and then pressing 'Enter' should produce the effect of adding multiple values to the database.

Additionally, the participants made some noteworthy comments to do with the 'add a case study feature' that didn't warrant a failing test case but are worth discussing. Participants noted that a lot of the fields for a case study (including the link) could be left blank when adding a case study, stating that there needs to be a minimum standard for the information needed to be able to add a case study to the database.



Ethics CS Portal

Add a Case Study

Case Study Title

Brief Overview

System name is different to the company name, input the name of the technology system this case study involves. If system name is not clear, please put N/A

System Name

For the following fields that can contain multiple inputs, please press 'Enter' to add a value in the field.

Operator(s)/Company(ies)

Technology(ies)

Figure 4.6: Add a case study, problem with multiple input fields, highlighted in red

4.4.1.2 Location of favourites

The third bullet point in section 4.3.1.5 discusses complexity of a user finding their favourites. Following these issues found by the HCI students, the favourites section was made easier to access by renaming the ‘Go to profile’ tab to ‘Favourites’. Participants however, still had trouble accessing their favourites and voiced their opinion that it would perhaps be clearer if their favourites could be accessed without having to click on the profile icon, by adding a star in the top right next to the profile icon.

4.4.1.3 General usability issues

The following is a list of general usability issues found by the participants.

- The ability to delete a case study from the database, a lecturer might make a mistake when adding a case study and will not have the ability to delete it.
- The ability to edit a case study. If other news articles are released discussing the topic or a lecturer wants to add more information they wouldn’t be able to do that without having the ability to edit a case study.
- Folksonomy style tagging. To improve the accuracy of the TFIDF search bar, it was noted that lecturers would like to add a feature where everyone had the ability to add extra tags relating to a particular case study. If more unique tags were added, it would increase the chance that a query from the search bar would return relevant results.
- No method to clear the search after typing.

- No tour/walkthrough when user first visits the website to show users how to use it.
- Styling of information on the dashboard means that there is lots of text on the left but a lot of empty space on the right.

4.4.2 Discussion of the findings

The results from this study were promising, six of the test cases had a 100% pass rate, three of them had a 66% pass rate and only one test case had a 33% pass rate. This would imply that the tool met the requirements set out by the participants overall.

Folksonomy style tagging would mean that anyone accessing the website can add general tags to a case study. This would enhance the TFIDF search discussed in section 4.2.1.2 as it would add more terms for the algorithm to search over.

The 'clear search' feature was one issue that arose during UAT. Adding this would improve usability, especially on mobile devices as the user could clear their search with one simple tap instead of having to copy the entire search and delete it in that way.

4.4.3 Fixing the issues

At this stage, work was directed toward fixing the issues found in the UAT. However, due to timing constraints, it was important to prioritise the test cases with a high failure rate first. This would ensure the website no longer failed any test cases, subsequently, other features could be added to enhance the tool.

Firstly, a minimum standard requirement was set when adding a case study, which meant that users would not be able to add the case study unless it contained essential information, like title, description, and news article links.

Secondly, the information displayed to users when adding a case study has been enhanced, as illustrated in figure 4.7. The user is now guided to simply press 'Enter' within the input field, which is a more intuitive approach.

Furthermore, the bookmark icon, as seen in figure 4.8 was preferred over a star icon as it more closely represents the notion of 'saving' or 'favouriting' a case study.

In addition to these improvements, other features were incorporated, including the option to delete a case study, tagging in the folksonomy style, and a button enabling mobile users to clear their search.

Other features that were added were the ability to delete a case study, folksonomy style tagging, and a button to clear search for mobile users.

4.5 Research question findings

In summary, the previous results can now be summarised in terms of the research questions set out in section 1.4.

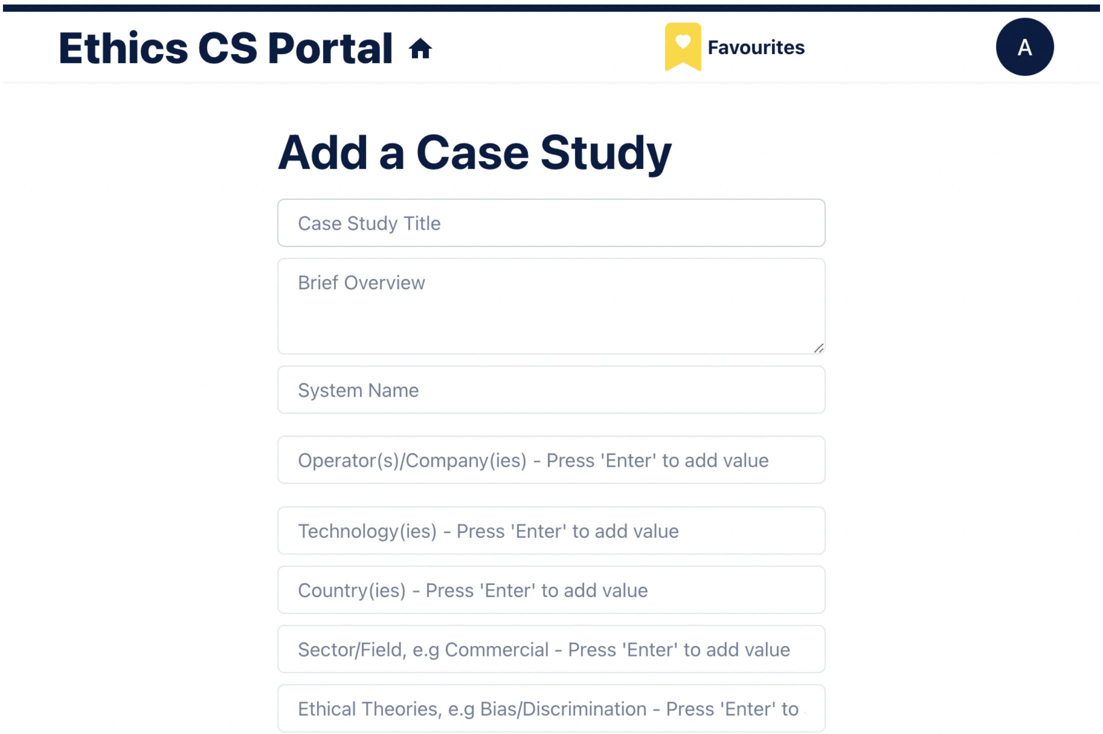

Ethics CS Portal
Favourites
A
Add a Case Study
Case Study Title
Brief Overview
System Name
Operator(s)/Company(ies) - Press 'Enter' to add value
Technology(ies) - Press 'Enter' to add value
Country(ies) - Press 'Enter' to add value
Sector/Field, e.g Commercial - Press 'Enter' to add value
Ethical Theories, e.g Bias/Discrimination - Press 'Enter' to

Figure 4.7: The ‘add a case study’ page final design.

1 - How have universities regarded the integration of computer ethics into CS degrees over the years?

As discussed in section 2.1, it was discovered that integrating computer ethics into CS over the years has received mixed views but is becoming more commonplace across CS degrees in recent times.

2 - What are the most effective methods of teaching computer ethics?

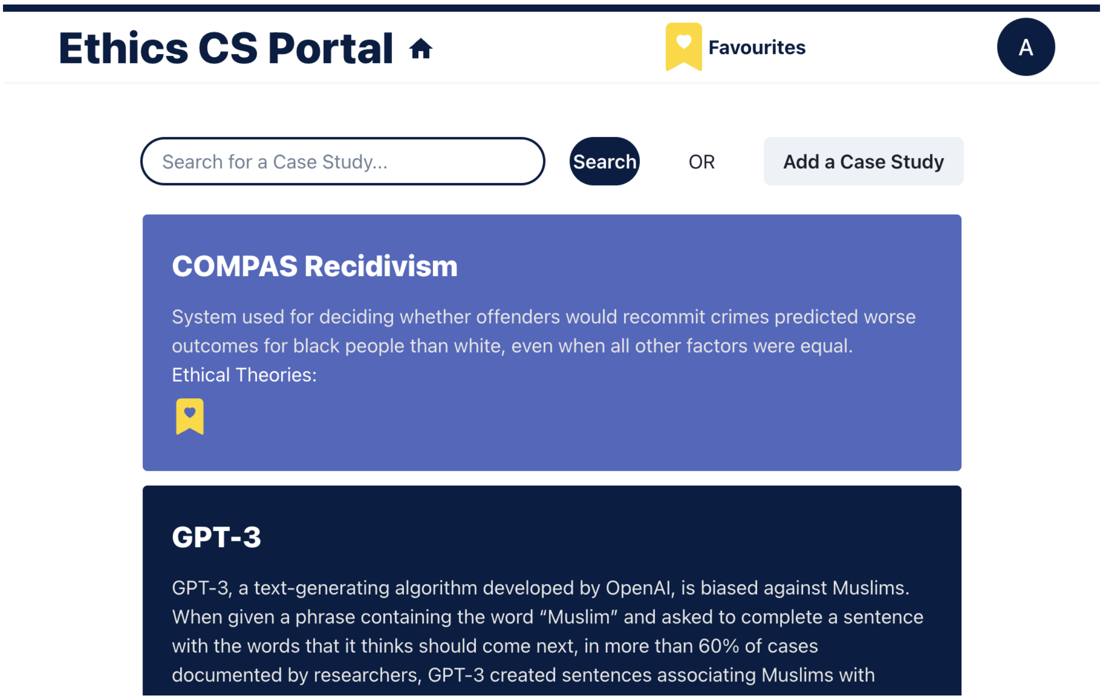
As highlighted in section 2.3 a discussion-based approach to teaching computer ethics was found to be the most effective way to deliver to students. Moreover, it discusses the effectiveness of case studies to initiate these discussions.

3 - What are the most effective methods of requirements gathering and design that could be used during the implementation of the proposed tool?

Methods of requirements gathering were discussed in section 2.6.2 and questionnaires were decided upon due to their simple online distribution characteristics. Subsequently, section 2.6.3.3 discusses the advantages of using a heuristic walkthrough when gathering design feedback as it incorporates two methods in one.

4 - Are there similar tools in this field and what are their strengths and weaknesses?

The section labeled 2.4 pertains to the findings gathered from investigating tools that were created for comparable objectives. Embedded EthiCS by Harvard was a tool of particular interest due to its user-friendly interface and search capabilities. However, it was noted that the method of organising content into modules could result in a sluggish



COMPAS Recidivism

System used for deciding whether offenders would recommit crimes predicted worse outcomes for black people than white, even when all other factors were equal.
Ethical Theories:

GPT-3

GPT-3, a text-generating algorithm developed by OpenAI, is biased against Muslims. When given a phrase containing the word "Muslim" and asked to complete a sentence with the words that it thinks should come next, in more than 60% of cases documented by researchers, GPT-3 created sentences associating Muslims with

Figure 4.8: The dashboard page, final design.

process for updating the database, resulting in outdated case studies. Secondly, the Ethics Education Library did not have an easy method of searching for specific case studies as they were structured in folders. Lastly, the AIAAIC's repository had a comprehensive list of case studies with lots of information for each one, however, the case studies were stored in a spreadsheet, and therefore searching for case studies was limited to the 'Find' command built into most operating systems, or by sorting the case studies by tags.

5 - What are the requirements of the tool?

A questionnaire was used (see section 3.3.1) to gather requirements and the results are discussed in section 4.2.2 where it was found that the main requirements were easy-to-add case studies, a way to store/bookmark case studies to make it easy to re-find them, and for it to be a community-driven database ensuring a comprehensive and up to date repository of case studies.

6 - How is the design perceived by human-computer interaction (HCI) students?

As expected, there were some usability issues in the third iteration highlighted in section 4.3, however, these were amended before carrying on with further testing.

7 - Is the final product a viable tool? Is it usable? And does it meet the initial requirements?

The evaluation confirmed that participants were pleased with the final tool as it did what was intended. The lecturers could add, browse, and store case studies for later viewing.

Chapter 5

Conclusions and further work

This chapter summarises the work that was carried out during the implementation of the Ethics CS Portal. It discusses the hurdles that were overcome, the limitations of the tool, and provides a brief overview of future work that could be carried out in this field. Finally, it provides some concluding remarks and summarises the core outcomes of this report.

5.1 Discussion

5.1.1 Difficulties encountered

During the implementation of the tool, there were some hurdles that needed to be overcome. The first issue was found when recruiting participants. A list of 39 potential participants across the university was collated and in total seven responded and completed the questionnaire, three of whom took part in the UAT at the closing stages of implementation, this works out at an 18% and 8% response rate respectively.

Secondly, the qualitative nature of the results obtained from the questionnaire in section 4.2.2 meant that it was challenging to group similar answers together due to textual ambiguity. For example, answers provided by two different participants might be referring to the same thing but look entirely different in textual form. Therefore, the analytical process was lengthy as the answers needed to be first summarised and grouped together to prioritise requirements.

Finally, the last issue was a more vague dilemma, however, it is worth noting. The problem was that as more features were added to the tool there were more opportunities for bugs, slow processing times, and website crashes to occur. It was therefore decided that the most important features would be given priority for the initial release of the tool. Other 'secondary' features, such as popularity score and tags to indicate which case studies have already been used in other courses were therefore omitted but could be added in future work.

5.1.2 Limitations of the tool

Because the tool was created through a community-driven approach (explained in section 4.2.2.2), it's probable that the lecturers themselves will contribute to the database. As discussed in section 2.1, the ethical information provided in each case study may not be entirely accurate, however, as the tool allows anyone to access it, users with advanced knowledge of philosophy can supplement the information by adding relevant details if they see fit.

Additionally, if the tool were to be implemented within the university's system, there would need to be a reassessment of the logistics involved in hosting the server. This would include the fact that the university has its own servers, and therefore, the password hashing process, and data privacy management would have to be carried out by the tool itself rather than being outsourced to an external provider (such as Firebase in this particular case). If the tool was hosted on the universities system, a permissions system could be added, whereby students could access the website with limited functionality, lecturers would be given the ability to add, delete and edit case studies and admins would have even more control over the website's design and usability for future versions.

5.2 Further work

The tool exists as a fully functioning website that meets the requirements set out by the participants. However, participants identified some additional features as set out below which could improve the overall effectiveness and usefulness of the tool in future versions.

5.2.1 Popularity score

Participants proposed the idea of adding a popularity score for each case study. The more lecturers that use a case study the higher the popularity score.

5.2.2 Tags for which courses have used a case study already

Participants also discussed the problem of not knowing which case studies their students have already worked on. Hence, a method of identifying which course the case study has been used in can be added to the website to mitigate this.

5.2.3 Editing a case study

The ability to edit a case study was not added to the tool due to timing constraints. This feature would be useful as it would allow multiple people to edit a case study and give their own interpretation of the information that it should contain.

5.2.4 Comments

The ability for lecturers to leave comments on case studies would expand the purpose of the tool as lecturers could share their experience of using the case study, and they could also propose and comment on class activities making it more of a social platform.

5.2.5 Expanding the use case of the tool

While the tool was primarily designed for educational purposes, it's important to recognise that it could also offer value in other fields. For instance, the Ethics CS Portal could serve as a repository for startups that are interested in learning about cases where similar technologies to theirs have resulted in adverse outcomes. These groups of people would then understand these potential implications and strive to mitigate them in whatever they decide to make. This was discussed as part of a meeting with the Chair of the Informatics Ethics Committee at the University of Edinburgh, who noted that this resource would be especially useful as a lot of people with questions regarding the potential ethical implications of their startup could be re-directed to the Ethics CS Portal for more information.

5.3 Concluding remarks

5.3.1 Core contributions

The following outcomes were found during the course of this research.

- Computer ethics has received mixed opinions from scholars in the past, however, in recent times more universities are incorporating discussions of computer ethics into their courses.
- Case studies are an effective pedagogical method for discussing ethics in CS.
- Similar tools exist on the web but either lack methods to search through case studies or have long creation times for each module.
- Requirements found from participants stated that the tool should be community driven, have a quick easy-to-add approach, and have a way to store/bookmark case studies.
- The tool was firstly sketched on paper, then designed on Proto.io and finally coded in React.
- Some usability bugs were found during the heuristic walkthrough. These were amended before the final evaluation.
- The overall feedback from participants from the final summative evaluation was positive with only a few test cases failing. These were again amended in the final product.

Various research was carried out to construct a foundation for the Ethics CS Portal. Firstly, universities changing views on the integration of computer ethics were investigated. Over the years computer ethics has received mixed opinions from scholars, however, in recent times more universities are incorporating discussions of computer ethics into courses, this uptake was observed from research from 62 universities across Europe [24]. Further background research delved into the most effective pedagogical methods for computer ethics. Many techniques were found to be successful according to various papers, including discussions using case studies, role-playing, and a software engineering project [15] [5]. To conclude the background research, websites with a similar use case to the proposed Ethics CS Portal were researched to identify their strengths and weaknesses. There were three websites that stood out, however, some flaws were noted, one website, namely Embedded EthiCS by Harvard was found to follow a module-based repository in which the information related to each module was extensive which makes the addition of materials to the database a long process which might not be able to keep up with the capricious nature of technology. Other resources had a flat database structure and also didn't include search functionality which made it increasingly difficult to search through the vast list of case studies.

Work was then directed to internal research and firstly a questionnaire was sent to a selected list of participants, these questions would gather requirements for the proposed tool. The main outcomes were a way to store/bookmark case studies, minimal effort in adding to the database, and for it to be community driven.

Following this, time was spent creating an initial mockup of the tool, using rough sketches. These were used as a basis to create prototypes using software for a more sophisticated design. The final stage of development of the tool was the implementation, which was coded in React, a Javascript library.

After an initial version of the tool had been developed, a heuristic walkthrough was carried out on human-computer interaction students and some flaws were found and amended before the final evaluation stage could take place. This final evaluation consisted of a user acceptance test where participants commented on whether the requirements set out in the initial questionnaire had been met. In general, the feedback from participants was very positive, most test cases passed with a 100% success rate and only a few of them failed, as seen in section 4.4. The tool was amended until all of the success criteria were met, and a fully functioning tool was built.

5.3.2 Conclusion

The Ethics CS Portal has achieved its goal of developing a tool to assist lecturers in selecting case studies for teaching computer ethics. Based on in-depth requirements gathering, see section 4.2.2, backed by effective HCI techniques, along with quantitative and qualitative analysis of the design and the tool's features see section 4.3 and 4.4, Ethics CS Portal has brought user needs to the forefront and created a useful resource for lecturers.

Bibliography

[1] Confessore, N. (2018). Cambridge analytica and facebook: The scandal and the fallout so far. *The New York Times*.

[2] Dix, A. (1970). *Human-Computer Interaction*. Springer US.

[3] Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on software engineering*, 21(2):146–162.

[4] Fiesler, C., Garrett, N., and Beard, N. (2020). What do we teach when we teach tech ethics? a syllabi analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 289–295, New York, NY, USA. Association for Computing Machinery.

[5] Gotterbarn, D. (1991). Ethical considerations in software engineering. *[1991 Proceedings] 13th International Conference on Software Engineering*. pages 266 – 274.

[6] Grosz, B. J., Grant, D. G., Vredenburgh, K., Behrends, J., Hu, L., Simmons, A., and Waldo, J. (2019). Embedded ethics: Integrating ethics across cs education. *Communications of the Acm*, 62(8):54–61.

[7] Hanington, B. M. and Martin, B. (2012). *Universal Methods of Design: 100 Ways To Research Complex Problems, develop innovative ideas, and Design Effective Solutions*. Rockport Publishers.

[8] HigherEdPartners (2022). The future of work: How universities can prepare students for an uncertain future.

[9] Hotten, R. (2015). Volkswagen: The scandal explained. *BBC News*.

[10] Johnson, D. (1994). Who should teach computer ethics and computers society? *SIGCAS Comput. Soc.*, 24(2):6–13.

[11] Kovac, J. (1996). Scientific ethics in chemical education *Journal of Chemical Education, v73 n10 p926-28.*

[12] Leung, H. K. and Wong, P. W. (1997). A study of user acceptance tests - *Software Quality Journal*.

[13] Lewis, C., Polson, P. G., Wharton, C., and Rieman, J. (1990). Testing a walk-through methodology for theory-based design of walk-up-and-use interfaces: *Proceedings of the SIGCHI conference on human factors in computing systems*.

[14] Macrina, F. L. and Munro, C. L. (1995). The case study approach to teaching scientific integrity in nursing and the biomedical sciences. *Journal of Professional Nursing*, 11(1):40–44.

[15] Miller, K. (1988). Integrating computer ethics into the computer science curriculum. *Computer Science Education*, 1(1):37–52.

[16] Moss, E. and Metcalf, J. (2019). The ethical dilemma at the heart of big tech companies. *Harvard Business Review*.

[17] Neilsen, J. (1994). 10 usability heuristics for user interface design. *Nielsen Norman Group*.

[18] Nielsen, J. (1995). How to conduct a heuristic evaluation. *Nielsen Norman Group*, 1(1):8.

[19] Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research - education and information technologies.

[20] QAA (2022). Subject benchmark statement: Computing.

[21] Quinn, M. J. (2006). On teaching computer ethics within a computer science department. *Science and Engineering Ethics vol. 12, pp 335–343*.

[22] Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

[23] Sears, A. (1997). Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human–Computer Interaction*, 9(3):213–234.

[24] Stavrakakis, I., Gordon, D., Tierney, B., Becevel, and et al. (2022). The teaching of computer ethics on computer science and related degree programmes. a european survey. *International Journal of Ethics Education 7, 101–129*.

Appendix A

Questionnaire

This questionnaire was given to participants as part of the requirements-gathering process for the Ethics CS Portal.

EthicsDB - Creating a Case Study Database for Teaching Computer Ethics

1. Describe how you would go about finding a case study to teach ethics in your course. (Before the implementation of my software) Do you have an Excel Spreadsheet that you use to keep URL links to a case study? Or do you browse the internet to find a trending news article that deals with Ethics in AI?

   Enter your answer

2. What are the positives to this approach?

   Enter your answer

3. What are the drawbacks of this approach?

   Enter your answer

4. When you have found a case study that you plan to use, do you write your own description of the case study? or do you look for an abstract somewhere else?

Enter your answer

5. Do you create your own materials to teach the module? or would it be nice to get some inspiration of possible tasks/talking points on the website itself

Enter your answer

6. How do you assess the class to see if they have achieved the learning outcomes?

Enter your answer

7. I am planning on adding a tags field in the database so that you can search for specific sectors of AI Ethics. For example if you teach Law, you will be able to find all case studies relevant to Law. Do you think you would find this useful?

Enter your answer

8. Another feature I was planing on implementing was being able to 'favourite' and 'save for later' so that you will have a profile of case studies that you might use, avoiding having to search for the case study again. Would you find this useful?

Enter your answer

9. Would a comment section on each case studies be useful? This could be used by lecturers to leave comments on how the lesson went and if the students enjoyed it.

Enter your answer

10. Are there any other features that you would like to see in the new case study database website?(General question, feel free to bullet point)

Enter your answer

Appendix B

Heuristic Walkthrough Template

This template was filled out during a Microsoft Teams call with former HCI students during the heuristic walkthrough design evaluation stage.

Heuristic Walkthrough – Ethics CS Portal

Explain this to Evaluator.

Ethics CS Portal is a website backed by a database to simplify the process of teaching CS/AI Ethics in classes. The database contains a list of case studies related to different ethical challenges faced with the development of new technology. Lecturers can find all these case studies on the website and can search for the case studies that relate to their field. For example, a lecturer of Law at the University of Edinburgh can input keywords like 'Law' & 'Justice System' into the search and will be presented with the cases studies relevant to their course. The lecturer can then click on the case study they would like to use and get more details and can follow URL links to news articles on the issue. Hopefully this will mitigate the tedious process of using search engines to filter through unrelated articles and provide a tool where lecturers can also keep track of the case studies they have already used.

Cognitive 'Walkthrough'

Tasks

1. From the welcome page, sign up for the website by creating a new account as a staff member.
2. From the case studies main page. Search for a case study related to Bias, GPT-3 and OpenAI.
3. Click on this case study to view more details and then access the press articles URL link, finally return to the case study page.
4. From the case study page go back and favourite Open AI's case study
5. Go to your profile to view your favourited case studies.
6. Add a case study to the database.

Questions to Ask at Each Stage

Will the user try and achieve the right outcome?

Will the user notice that the correct action is available to them?

Will the user associate the correct action with the outcome they expect to achieve?

If the correct action is performed, will the user see that progress is being made towards their intended outcome?

Severity Ratings

- 0 – Not a problem – Evaluator thinks that this is not a usability problem at all.
- 1 – Cosmetic problem only – Need not be fixed unless extra time is available on project
- 2 – Minor usability problem – Fixing this should be given low priority
- 3 – Major usability problem – Important to fix, so should be given high priority
- 4 – Usability catastrophe – imperative to fix this before product can be released

Evaluation

From the welcome page, sign up for the website by creating a new account as a staff member.

Step	Will the user try and achieve the right outcome?	Will the user notice that the correct action is available to them?	Will the user associate the correct action with the outcome they expect to achieve?	If the correct action is performed, will the user see that progress is being made towards their intended outcome?	Summary of Problem + Severity rating
Click the Sign-Up Button					
fill in details and check the 'I am part of the staff' checkbox					
Click register					

From the case studies main page. Search for a case study related to Bias, GPT-3 and OpenAI.

Step	Will the user try and achieve	Will the user notice that the	Will the user associate the	If the correct action is performed,	Summary of Problem +

	the right outcome?	correct action is available to them?	correct action with the outcome they expect to achieve?	will the user see that progress is being made towards their intended outcome?	Severity rating
Input words like 'openai gpt bias' into the search bar at the top of the page					
Press Search					

Click on this case study to view more details and then access the press articles URL link, finally return to the case study page.

Step	Will the user try and achieve the right outcome?	Will the user notice that the correct action is available to them?	Will the user associate the correct action with the outcome they expect to achieve?	If the correct action is performed, will the user see that progress is being made towards their intended outcome?	Summary of Problem + Severity rating
Click on the white button for the cases study					
Find the Press					

Links section					
Press the URL link					
Finally press the back button in your browser.					

From the case study page go back and favourite Open AI's case study

Step	Will the user try and achieve the right outcome?	Will the user notice that the correct action is available to them?	Will the user associate the correct action with the outcome they expect to achieve?	If the correct action is performed, will the user see that progress is being made towards their intended outcome?	Summary of Problem + Severity rating
Input words like Click the grey star					

Go to your profile to view your favourited case studies.

Step	Will the user try and achieve the right outcome?	Will the user notice that the correct action is available to them?	Will the user associate the correct action with the outcome they expect to achieve?	If the correct action is performed, will the user see that progress is being made towards their	Summary of Problem + Severity rating

				intended outcome?	
In the top right click on the profile avatar					
Click go to profile in the dropdown menu.					

Add a case study to the database.

Step	Will the user try and achieve the right outcome?	Will the user notice that the correct action is available to them?	Will the user associate the correct action with the outcome they expect to achieve?	If the correct action is performed, will the user see that progress is being made towards their intended outcome?	Summary of Problem + Severity rating
In the top right click on the profile avatar					
click on add a case study					
fill in all the fields					
click the ‘add a case					

study' button.					

Heuristic Evaluation

Visibility of system status
Match between system and the real world
User control and freedom
Consistency and standards
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help users recognize, diagnose, and recover from errors.
Help and documentation.

HE Number: 1
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 2
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 3
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 4
Name:
Heuristic:

Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 5
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 6
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 7
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

HE Number: 7
Name:
Heuristic:
Interface Aspect:
Explanation:
Severity Rating:
Possible solution and/or trade-offs:

Appendix C

User Acceptance Testing Template

This template was filled out during a Microsoft Team call or filled out in their own time by each participant who agreed to carry out the user acceptance testing final evaluation.

Ethics CS Portal – User Acceptance Testing

Please use this link to access the website: https://ethicsdb2.web.app/

Hi all,

Thanks for agreeing to participate in the UAT for Ethics CS Portal. Below is a brief description of the software if you needed a quick refresher. Don't worry if you didn't answer the previous Microsoft Form requirements gathering questions and if this is your first-time hearing about my project the information below should help.

The Idea

Ethics CS Portal is a website backed by a database to simplify the process of teaching CS/AI Ethics in classes. The database contains a list of case studies related to different ethical challenges faced with the development of new technology. Lecturers can find all these case studies on the website and can search for the case studies that relate to their field. For example, a lecturer of Law at the University of Edinburgh can input keywords like 'Law' & 'Justice System' into the search and will be presented with the cases studies relevant to their course. The lecturer can then click on the case study they would like to use and get more details and can follow URL links to news articles on the issue. Hopefully this will mitigate the tedious process of using search engines to filter through unrelated articles and provide a tool where lecturers can also keep track of the case studies they have already used.

UAT

Question: Did you complete the requirements gathering questionnaire on Microsoft Forms? Please answer here: YES/NO

The Acceptance Criteria

The following requirements should be met if the website can be considered acceptable.

- Users should be able to sign up/login to the application.
- Users should be able to browse case studies on the database.
- Users should be able to search for keywords and be presented with results of case studies related to their search.
- Users should be able to obtain more information about the case study easily.
- Users should be able to access news articles related to the case study easily.
- Users should be able to favourite/'save for later' case studies of interest.
- Users should be able to view these favourited case studies easily.
- Lecturers should be able to add a case study to the database.
- The overall usability of the application should not impede a user's ability to complete all these tasks.

Test Cases

Case 1: Can you sign up for the website. (Please use dummy email and password, all passwords are encrypted with Google's firebase encryption service)
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 2: Can you Login to the website with the details you just registered with?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 3: Can you get more details of a case study you are interested in?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 4: Can you favourite a case study.
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 5: Can you view your favourites easily?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 6: Can you search for a case study?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 7: Are the results relevant to your search?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 8: Is the navigation on the application intuitive and easy?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 9: Is the website fast (no long waiting times)?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Case 10: Can you add a case study to the database?
Pass/Fail – Please give your answer here: _______
(If Fail), please give a quick description of why it failed here:
Any other comments:

Please answer the following questions too.

How would you rate the usability of the application on a scale of 1 -10 (1 = Really easy to use, 10 = Really difficult to use)

1 2 3 4 5 6 7 8 9 10

Any other comments?

Appendix D

Online research participant information sheet (PIS) and consent form

Due to all of the data collection in this report being completely online, the combined PIS and consent form could be given to the participants.

Two participant information sheets are attached. The first one was given to the participants of the questionnaire and the second related to the heuristic walkthrough and user acceptance testing

Participant Information Sheet

Project title:	EthicsDB
Principal investigator:	James Garforth
Researcher collecting data:	Oscar Alberigo
Funder (if applicable):	N/A

This study was certified according to the Informatics Research Ethics Process, RT number 515722 Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

Oscar Alberigo (s1861004)

What is the purpose of the study?

To gather some requirements/features for my website that I will be implementing (An Ethics Case Study Database for lecturers to teach in classes)

Why have I been asked to take part?

Lecturers across various departments at the University of Edinburgh widely use case studies to teach CS ethics modules in their classes, I would like to see what the process is right now for finding the case studies and distributing them to the class

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, without giving a reason. Your rights will not be affected. If you wish to withdraw, contact the PI. We will stop using your data in any publications or presentations submitted after you have withdrawn consent. However, we will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

The questions that will be asked will be to do with how you go about finding cases studies related to the course that you teach. And how you use them in your class to teach ethics.

The data will be collected through an anonymised online questionnaire (most likely google forms or Microsoft equivalent)

It will take around 10-15 minutes to complete

Questions are written answers, there will be around 8 questions

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

The benefits of your co-operation will mean that the website might include the features you would like to see implemented and then, once the website is finished, lecturers can use it to find case studies much easier than it is now.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of 4 years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team, myself (Oscar Alberigo, s1861004) and James Garforth my supervisor

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Oscar Alberigo, s1861004@ed.ac.uk
If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on http://web.inf.ed.ac.uk/infweb/research/study-updates.

Consent

By proceeding with the study, I agree to all of the following statements:

- I have read and understood the above information.
- I understand that my participation is voluntary, and I can withdraw at any time.
- I consent to my anonymised data being used in academic publications and presentations.
- I allow my data to be used in future ethically approved research.

Take me to the survey.

Participant Information Sheet

Project title:	EthicsDB
Principal investigator:	James Garforth
Researcher collecting data:	Oscar Alberigo

This study was certified according to the Informatics Research Ethics Process, RT number 257244 Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

Oscar Alberigo, James Garforth

What is the purpose of the study?

To gather feedback on the front-end design of my Ethics Case Study application

Why have I been asked to take part?

Students who have previously taken the Human Computer Interactions (HCI) course. If possible, try and get some lecturers/ HCI specialist opinions too.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, without giving a reason. Your rights will not be affected. If you wish to withdraw, contact the PI. We will stop using your data in any publications or presentations submitted after you have withdrawn consent. However, we will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

You will be invited to a Teams call where I will take you through a cognitive walkthrough where I will gather you feedback on the design of the application

It will be an discussion based interview

The video/audio will be recorded for analytical purposes (so that I can remember everything that was said)

The Teams call will take about 30 minutes and it will be a one off call.

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

No

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a minimum of 2 years.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team, Oscar Alberigo and James Garforth

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

For general information about how we use your data, go to: edin.ac/privacy-research

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Oscar Alberigo (s1861004)

If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on http://web.inf.ed.ac.uk/infweb/research/study-updates.

Consent

By proceeding with the study, I agree to all of the following statements:

- I have read and understood the above information.
- I understand that my participation is voluntary, and I can withdraw at any time.
- I consent to my anonymised data being used in academic publications and presentations.
- I allow my data to be used in future ethically approved research.