SocialEyes: Enhancing Social Media Collaboration for Disabled Users, with Eye Tracking

140012021 Christopher Alexander Fleming

University of
St Andrews

17th May 2019

Supervisor: **Ruth Letham**

1 Abstract

Extensive research has gone into demonstrating how social media and virtual environments can help disabled users. The field extends across from medicine and psychology to Computing. A key focus to allowing the benefits is how the users interact with the applications, with eye tracking being one of the key components. However, eye tracking has many issues with it, that become worse due to how social media sites change and alter their layouts, leading to slower and less accurate interactions.

The aim of the project is to create a social media collaboration tool designed to be used by fully disabled users with eye tracking being the sole input. The application incorporates a variety of designs and techniques offering different methods of making inputs quicker and more accurate, through filtering news feeds and menus used for eye tracking inputs. This allows users to be more involved in social environments, which can help with their physical and mental well-being.

The application called SocialEyes can tag items from multiple social media sites showing where they originated from and also what topic the posts are about, which in turn allows the users to use a filter menu to remove and add topics and social media sites they wish to see or not see. A common set of social media features appears in SocialEyes, allowing users to interact with the central news feed. The menus in the system were customised to help the handling the eye tracking input. SocialEyes was compared to a current alternative in a user study focusing on speed and accuracy of inputs, with significant results being reported.

2 Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 17,136 words long, including project specification and plan.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bonafide library or research worker, and to be made available on the WorldWide Web. I retain the copyright in this work.

Contents

List of Figures

3 Introduction

The main aim of this project was to create a social media collaboration tool, with eye tracking as its only input, to be used by physically disabled users which can help them interact with others.

The use of eye tracking as an alternative input allows disabled users to interact with systems in a way that would not be ordinarily achievable. In 1987 Ware and Mikaelian published a paper that evaluated an eye tracker as a form of computer input. They state about a user dwelling on a location for a set period of time in order to make a selection. This is necessary for disabled users, where eye movements may be the only body movement the user has control over[39].

In various experiments published, eye tracking for disabled users can allow them to perform the same actions as an abled body user, which improves the day to day life of a disabled user. The eye typing system Dasher is an example of a new system designed to bring eye tracking level with other forms of input, with it producing a text entry rate of 34 words per minute[38].

There is a growing issue around physically disabled users being able to use general computing interfaces. Social media platforms are an example of systems that are currently changing. The platforms persistently change, with new features continually being added and removed. The changing layout of a platform can make it challenging for a disabled user to become familiar with the system allowing them to obtain the full benefits of using an eye tracker.

The project involved creating a social media collaboration tool which improves the experience for disabled users that need to interact with systems using eye tracking. A Tobii EyeX tracker was implemented with SocialEyes, allowing the user to dwell on different menus designed to aid in the interactions a disabled user performed.

The project aimed to demonstrate this through the use of a user study comparing the application created for the project called SocialEyes and the current web-based alternative. The user study compared the speed and accuracy of completing a common set of social media features, with the results checked for significance through a series of statistical tests.

The secondary aims of the project included allowing the user to filter their news feed items, having the system adapt to their requirements and allow the users to decide what social media features they wish to use.

4 Objectives

The objectives of this project as set out at the beginning of the project after discussion with the project's supervisor (January 2019):

4.1 Primary Objectives

P1: Conduct research identifying a feature set common amongst social media platforms.

P2: Design and Prototype a system for managing multiple social media platforms using eye tracking as the input.

P3: Create a final application using the identified feature set allowing users to interact with items/posts.

P4: Conduct a study comparing the created system against other similar systems, analysing the quantitative and qualitative results.

4.2 Secondary Objectives

S1: Allow users to filter items in the central feed.

S2: Allow users to customise which features from the feature set they want to use.

S3: Have the UI adapt to filtered/customisable changes the user chooses.

5 Requirements

5.1 Social Media Feature Sets

One of the objectives of the project was to identify a common set of social media features which would be collated and analysed to decide what features were key for the application and could be incorporated into the social media collaboration tool.

Early forms of social media often contained a feature that allowed users to create a piece of information and placing it in a central location. This was an early form of *posting*. Tom Truscott and Jim Ellis created Usenet, a worldwide discussion system that allowed internet users to post public messages, with Bruce and Susan Abelson 20 years later founding Open Diary, where online diary writers were brought together into one community[10]. Both of these social media platforms are examples of early adoption of *posting*. The idea was continued with most modern sites, such as Facebook, Twitter, Reddit etc, still using

the concept today.

Most modern social media sites provide additional non text based feedback systems. For example Youtube's original five-star rating. The issue with such a system is that when people like a video they often give it five stars, meaning the majority of videos end up with five-star ratings[41]. A simpler a more widely used form is the *like* feature. Many platforms have their form of the like feature as "it has positive impact as it indicates that the user found content interesting, useful or worth considering"[21]. Different platforms extended the concept. Facebook introduced *reactions*, with the purpose of these options allowing users to be less ambiguous in their responses[21]. Reddit's downvote system which reflects in a user's karma[24] and Pinterest use pinned posts to show interest.

Another social media feature currently used is *share.*"By sharing respective content via internet-based business social networks, a user suggests not only that the respective content, concept or idea is interesting, useful or worth considering, but also worth promoting further within the community."[21]. Sharing is a common feature in all social media sites so it was a useful feature for the tool in development.

Based on the above features identified, the common features identified for SocialEyes were: Creating a post, commenting, liking and sharing. SocialEyes would also have a central feed where the identified features were performed. Features such as reacting and down voting, were deemed to be future extensions to the system.

5.2 System Requirements

This section details the requirements list developed in the initial requirements capture phase of the project. They were used to assess whether the project was remaining in scope. The MoSCoW method was used to produce a list of requirements for the system. *Must have* is something the project guarantees to deliver, *Should have* is important but not vital to the system, Could have is desirable to the system and *Won't have* is something that will be useful for the system to have in future. But will not be implemented in this project.

ID	Requirements	Priority	Dependencies
1	User **must** be able to attach their Twitter account to the system	High	None
2	User **must** be able to attach their Reddit account to the system	High	None
3	User **must** be able to see a combined feed of their social media accounts	High	1,2
4	User **must** be able to *Like* a post in the central feed	High	3
5	User **must** be able to *Comment* on a post in the central feed	High	3
6	User **must** be able to *Share* a post in the central feed	High	3
7	User **must** be able to *Create* a post in the central feed	High	3
8	User **should** be able to identify which social media site a post is from	Medium	3
9	User **should** be able to identify the category of a post	Medium	3
10	User **should** be able to filter their central news feed	Medium	8,9
11	User **should** be able to change the dwell time	Medium	None
12	User **should** be able to customise the selection menu	Medium	3,4,5
13	User **should** be able to refresh their central feed	Medium	3
14	There **could** be a function that saves current system settings	Low	None
15	There **could** be a function adds new tag types to feed items	Low	None
16	There **won't** be a method for adding and removing social media sites	N/A	None
17	System **won't** be used on systems other than Windows	N/A	None
18	System **must** use eye tracking as sole input	High	None

Each of these requirements is linked to one or more of the project objectives.

- By implementing requirements 1-3, objective P2 will be satisfied.

- By implementing requirements 4 - 7 objective P3 will be satisfied.
- By implementing requirements 8 - 10 objective S1 will be satisfied.
- By implementing requirements 11 - 13 objective S2 will be satisfied.
- By implementing requirements 11 - 14 objective S2 will be satisfied.

6 Context Survey

This section describes the current field of eye tracking, computing for disabled users and benefits of social media for disabled users. It will also analyse the eye tracking projects that have been published, looking at possible improvements in the systems.

6.1 Eye Tracking

Eye-tracking aims to detect what a user is gazing at on the screen and applications use this information to allow the user to interact with computers[20]. These different behaviours that occur inside eye tracking input help understanding when linking to the main issues of eye tracking that research methods investigate. The common issues that affect eye tracking are[9];

- Hardware problems including the usability and cost of eye trackers.
- The inherent inaccuracy of eye tracking.
- The Midas Touch problem.

6.1.1 Hardware Problems in Eye Tracking

The accuracy of the various pieces of hardware that specialise in eye tracking are summarised by Gibaldi et al.[5]. At the highest end of the cost scale the accuracy is still up to 0.8 degrees (see figure 1) , therefore, gaze targets areas need to be large enough to reduce any error that is purely from the eye tracking hardware.

Table 3 Commercial Eye Tracker Comparison

Eye Tracker	Accuracy [deg]	Precision [deg]	Sampling Rate [Hz]	Latency [ms]	Price Point [$]
EyeX	0.5-1	0.25	55	<50	~ 100
EyeTribe	0.5-1	0.1	30-60	<20	~ 100
GP3	0.5-1	0.1	60	<50	<1000
myGaze	0.5	0.1	30	<50	<5000
SMI-REDm	0.5	0.1	60-120	<20	<25000
ViewPoint	0.25-1	0.15	90-220-400	<10	<25000
EyeLink 1000	0.25-0.5	0.01	250-500-1000-2000	<10	>25000
Tobii TX300	0.3-0.8	0.1	60-120-250-300	<10	>25000

Figure 1: Performance of Different eye trackers[5].

Cheung and Peng investigated the accuracy of using a web camera as an eye tracker. The advantage of using a webcam over various infrared eye trackers is that the users can wear glasses, due to the lack of reflection generated with the webcam.

The issue surrounding the quality of the first problem is no longer an issue with "an accuracy of 1.5 degrees was obtained while keeping the hardware cost of the device below one-hundred euros"[13].

6.1.2 Inaccuracy of eye tracking

Another solution that was investigated to solve accuracy issues in eye tracking is by using zooming. The system EyePoint follows this concept using a combination of eye gaze and keyboard triggers. It uses a two-step progressive refinement process fluidly stitched together in a look-press-look-release action, which compensates for accuracy limitations of eye trackers[12]. However, the problem with the systems that include various extra techniques is that they are created for non-disabled users such as in EyePoint. If a system was developed with disabled users the target, extra steps are required to ensure accuracy which directly affects the speed of the system.

6.1.3 The Midas Touch

The most common and hardest problem that faces eye tracking systems is the concept of the Midas Touch. Velichkovsky et al. define the Midas Touch as "how to differentiate 'attentive' saccades with the intended goal of communication from the lower level eye movements that are just random or provoked by external stimulation."

Eyes are an *always-on* device, an organ that is always conditioned to be looking at various objects rather than being used as an input method to a computer. The most common method to overcome the issue when no additional

input device is available is to increase the *dwell time*, which is the period of time a user gazes at a fixed point on the screen to make a selection. Increasing the dwell time ensures that no unintended action will occur. Long dwell times can be detrimental as they can be "fatiguing and can result in the gaze point moving off the intended target before the end of the dwell period, which leads to slow, effortful interaction"[9].

6.1.4 Overcoming Eye Tracking Issues

Zhang, Yao and Cai look into efficient eye typing by using 9-directional gaze estimation in an attempt to mitigate the possible reduced quality the eye tracking, along with reducing the calibration time required[42]. The application they developed was designed for disabled people to use in everyday scenarios and not in a lab like environment. Instead of using standard entry techniques they propose separating human gaze into nine directions.

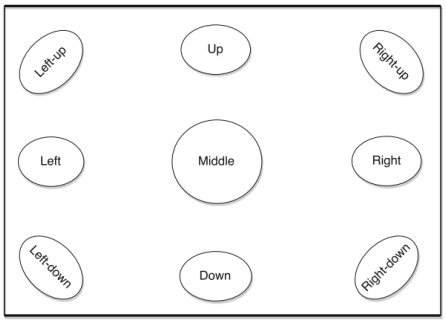


Figure 2: Corresponding regions on the screen of the 9 directions.

The advantage of splitting the screen into nine sections (see figure 2) as it leads to a range for each direction of between 30 and 70 degrees, counteracting the accuracy issues of eye tracking. The nine sections are based on the classic 9-key T9 input method keypad used on bar phones. Input is handled by pointing at the letters by gaze and selecting letters with voluntary blinks[42]. By having fewer keys on the screen it gives the users a larger target area for making a selection.

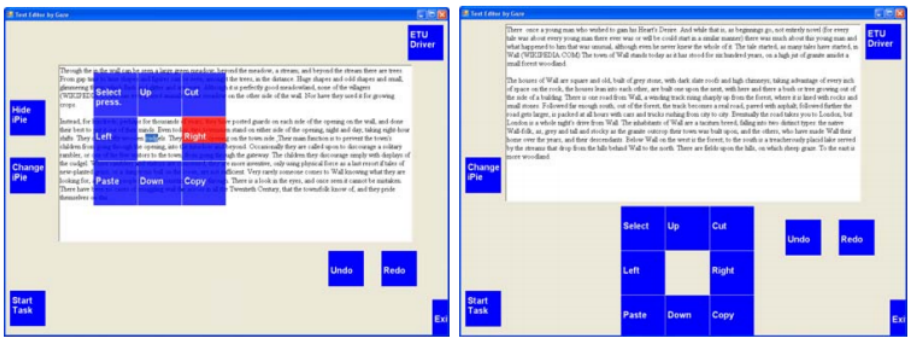


Figure 3: Dynamic and Static menus created by Majaranta et al.[14]

Another example of an application overcoming eye tracking issues for disabled users was using a dynamic pie menu[14]. The pie menu appears at the point of the user's focus when the user fixates on a piece of text for predefined dwell time. The user sees the text they wish to edit through a porthole in the centre of the menu, which can be copying the text, making it bold or italics etc. They also created a static menu, positioned at the bottom of the screen, where the same options could be selected from (see figure 3).

In an experiment comparing the menu systems, Majaranta et al. show that the dynamic menu has an average performance time of 38 seconds compared to the 47 seconds of the static menu system when performing simple tasks. The decreased time and being able to see the action they are performing increases the usability for disabled users.

Urbina and Huckauf look into addressing the Midas Touch problem by adopting a typing interface based on hierarchical pie menus. Several different versions of the menu were produced on how the text entry would be selected but all were based on the same menu structure.

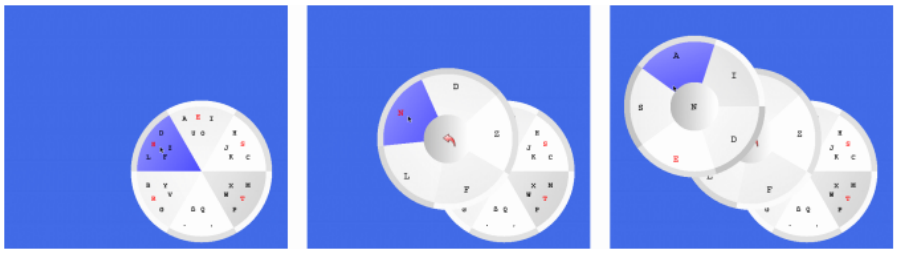


Figure 4: Process of hierarchical menu structure by Urbina and Huckauf[37].

At the highest level, there were multiple options to select from, which then filters down into fewer options, while having the original menu still present on the screen (see figure 4). The process of making a selection involves moving over a border, instead of using purely dwell time."The use of selection borders not only allows the user to inspect the pie as long he/she needs to, but also to perform fast multiple selections with saccades that follow each other"[37]. 8 out of 9 participates preferred selection borders compared to dwell time, with every user making more selections with selection borders than dwell time.

The methods mentioned overcame the Midas Touch problem using a similar method to how the accuracy issues were overcame. They used keyboard and mouse as methods of confirming selections, which is not an available option to fully disabled users.

6.2 Disabled Users Interaction Benefits

People that have a disability can often feel separate from society, but computers can offer a way to reduce this feeling. Virtual communities being usable through gaze is of great relevance to severely disabled people as it can enable them to be in the community on a similar basis to able-bodied participants[9]. By using games such as Second life [30] and Minecraft[16], users with disabilities say it can help them fight their diseases, live with their disabilities easier and even sometimes begin to recover. They find a central area where they can find people with similar issues to them helps and researchers have started to appreciate what kind of impact a virtual world can have[4].

The use of this style of Virtual Reality (VR) worlds also aids the mental health side of the user. EMMA (Engaging Media for Mental Health Applications) was a project funded by the European Union to create a virtual world to aid in the recovery of its users[4]. By allowing the patients a way to communicate it can aid both their physical and mental well being. The easiest way for people to communicate now is on social media sites. However, social media

sites are not designed for fully disabled users.

Social Media can also help form an excellent basis to make communication easier for disabled users. In Disability and Social Media, Ellis and Goggin open with "Social Media can break down barriers and change lives. For example, a hearing impaired child can now share stories with their friends via Skype, and a visually impaired user can read about his brother's trip"[2]. Social Media open new channels for users to be able to be a bigger part of different communities even if they live in remote communities themselves[29]. The use of Social Media can help disabled users, but they first have to be able to use it.

7 Ethics

An ethics form that covers the content of the module was submitted. Early in the project, people would compare different designs and give their views on them, with the feedback incorporated into the final design of the application. The form is in appendix A. (Approved by CS12476).

The proposal of the project included the concept of a user study taking place comparing the time and accuracy of SocialEyes and a current alternative, with the users finally answering a questionnaire. The ethics for the module does not cover the proposed user study, so a full ethics application was submitted. The form is in appendix B. (Approved by CS14257).

8 Design

After defining and producing the requirements and the objectives, a clear path for the system being created could then be incorporated into the prototyping process used during the design phase of Human Computer Interaction projects. During this phase, the design space of different possibilities were explored.

The process starts with low fidelity design techniques, which are used in the analysis phase of product development. Low-fidelity prototypes can be created quickly, efficiently and at a low cost to developers. They form the first step in proposing fundamental design approaches for the user interface. These prototypes can be demonstrated, with feedback then given back on them allowing the designer to re-evaluate as they continue towards high-fidelity processes.

8.1 Distinct areas of SocialEyes

There were five distinct areas that formed the basis for the design of SocialEyes;

- A central area for viewing posts.

- A method to filter the posts in the central area.
- A method to filter the actions the user could perform.
- A method to customise aspects of SocialEyes, such as dwell time.
- An area for creating posts.

The different areas were based on the requirements that were gathered in section 5.

8.2 Paper Prototype

The initial low-fidelity strategy selected was paper prototyping. Two designs were prototyped. The first (prototype A) followed an email-style layout. The Second (prototype B) used radial-style menus similar to the menus discussed in section 6.1. Both of the initial paper prototypes were presented to five people who represented the customer in the design process. They gave feedback on both of the designs, allowing another iteration of designs that were closer in resemblance to the final design.

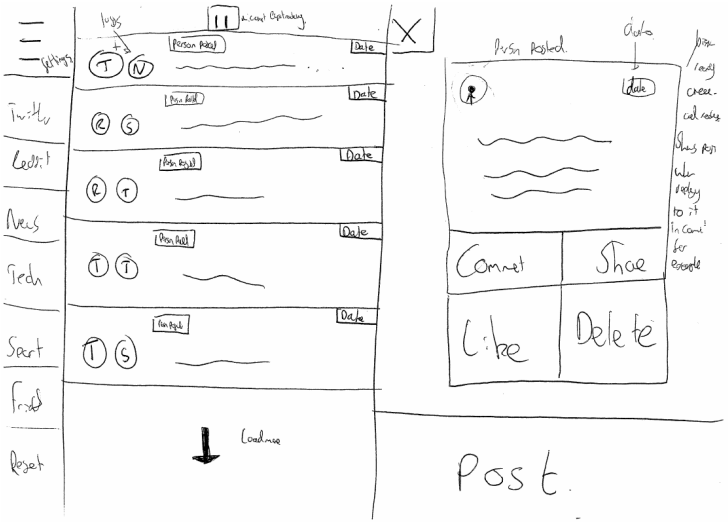


Figure 5: prototype A: Initial design of the system following an email style layout.

In figure 5, prototype A, an email style client was proposed where an item would be selected, creating a pop-up that would be the size of the screen. Each of the items in the email style feed would have circles with letters inside them to indicate what site the post was from and what kind of post it would be. The initial feedback on this design was that the small letter icons could be hard to identify. Additionally the colour coordination could, along with the sidebar be

used to filter the items.

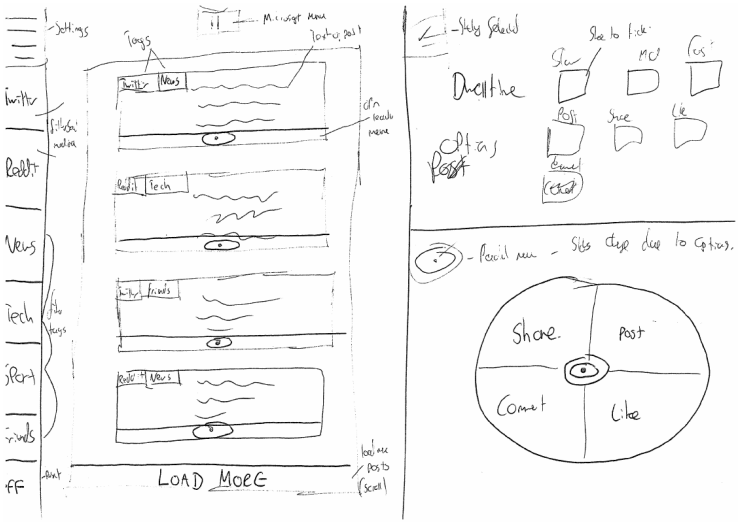


Figure 6: prototype B: Initial design of the system with central feed and radial menus.

In figure 6, prototype B was created containing a central feed with a larger area. There would no longer be a screen sized pop out, with each item being large inside the central news feed. A radial menu was investigated instead of having a grid-based system to select social media features. The feedback on the design was surrounding the *load more* and the radial menu. The users stated it could end up overloading the central feed and that the radial menu could be an unusual input technique, with users adapting to the grid system more quickly. This feedback marked the end of the first iteration, and the designs were altered to adapt to the feedback given. Prototype C was developed as a result.

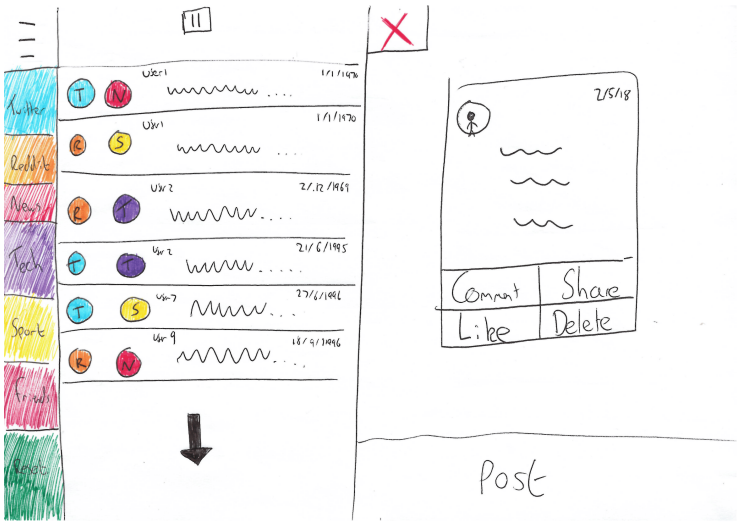


Figure 7: prototype C: Redesign of the system following an email style layou.t

The feedback was more positive with the colour being introduced on the email style layout in figure 7, allowing users to quickly identify the posts, even with minimal text on show, and the user feedback suggested that this allowed many types of posts to be seen, but not enough of a post would be visible to know if it was worth looking into.

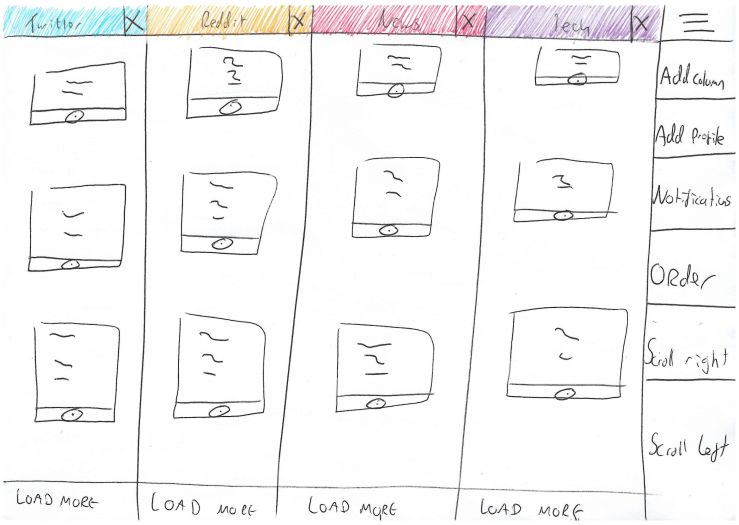


Figure 8: prototype D: Redesign of the system, changing central feed to multiple feeds.

Feedback on figure 8 focused on the column style of the multiple news feeds. Users raised concerns that the size of the areas selected with the eye tracker would be too small and mistakes could quickly occur, which echos the accuracy issues mentioned in section 6. However, they were much happier with the amount of information visible in a post. Therefore one final paper prototype E was produced to incorporate the feedback from the different designs.

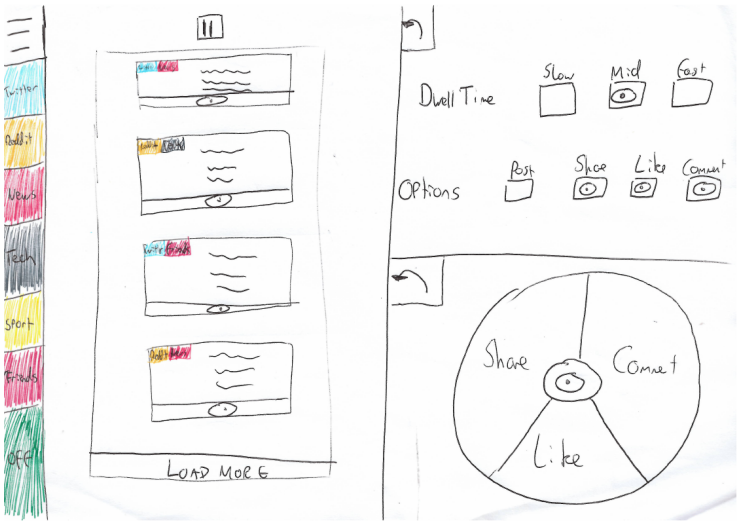


Figure 9: prototype E: Initial design of the system with central feed and radial menus.

Figure 9 shows the final paper prototype E. The colour identifies the type of post and which social media site it originates. The news feed structure decided on was the central news feed, with the concept of scrolling remembered. It was important for users to be able to see the post before they clicked on it, to save time for the users. The radial menu would also form the input for the system. After seeing the findings in section 6 users, agreed the design would make for a quicker and more accurate system.

8.3 Storyboard

The next stage of the design phase was to storyboard the final paper prototype. Storyboards are a "series of static windows or menus that can be rapidly generated and displayed, either singly or in a storyboard presentation"[28]. Storyboards have been used in visual storytelling media, such as film and television for years and the concept is similar in the design process, allowing multiple benefits. It allows real-world contexts to contribute to the design process, which means the designer, in this case, thinking of the issues that come with an eye tracking application compared to a system with standard input methods. It also allows designers to remember that the designs need to flow into each other and they are not isolated systems, stopping the end experience being detrimental.

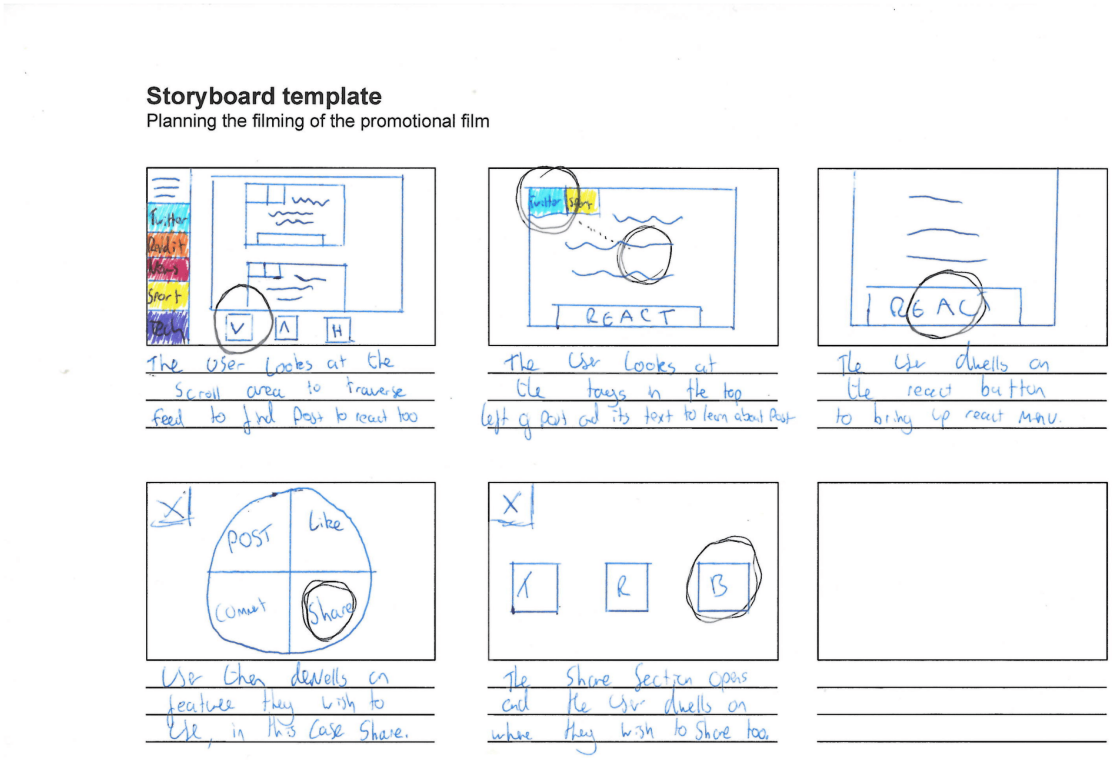


Figure 10: Storyboard dictating the scenario of a user sharing a post.

The storyboard in figure 10 shows a user wanting to share a particular post in the feed. The grey circle on the different scenes shows where a user's gaze would be. The tags on posts would need to stand out enough for a user to identify quickly, while not being accidentally selected, with the same being true for the text on each post.

The idea of changing the dwell time came during the requirements stages, but the specific periods were then thought of here, which would ultimately decide speed and accuracy the system could achieve. The storyboard would be similar for the different social media features identified in Section 5.1, so only one storyboard was produced, and it would form the basis of the high fidelity stage of the design phase.

8.4 Mock Application

When designers are working with high-fidelity prototyping, there are two directions they can take which are vertical and horizontal prototyping. In Vertical prototyping the number of features is lower, which results in a narrow system, that includes in-depth functionality. Vertical prototypes allow users to perform and test some real tasks[23]. While horizontal prototyping is where the functionality implemented is lower, so that the result covers the entire user interface, without the underlying functionality[23]. Horizontal prototyping allows the users to feel the entire interface, seeing how it flows together. The issue with vertical prototyping is that it results in real tests being run and analysed, causing the process to be significantly longer than horizontal prototyping and due to the time restrictions on the project it was not a viable option.

Many specialised tools exist to allow quick horizontal prototyping, including Mockplus[17]. The application contains different environments that can mock applications which allow everything being created to be the same scale it would be in the final application, an advantage that is not possible during the low-fidelity section of the design process. The technology used during the implementation will be explained more in section 9.1.1. However, the concept of UWP is that the graphics side of the application uses drag and drop, with more exact functionality introduced via code, therefore the mocked application would form a design very close to the final solution.

With the mock application under design, the eight golden rules in user interface design by Ben Shneiderman were then used to help shape the design[31].

One of the significant issues for disabled users, when they are using social media sites, is the tendency for the sites to change. Sites continually release updates, changing the look and feel of the websites. This makes it hard for disabled users due to the fact they use similarity to traverse the site, therefore one of the goals of the system was to allow the user to customise the application to what they want to achieve with the application, which supports Shneiderman's

golden rule of supporting internal locus of control. This concept is that the operators of an interface feel they are in charge of the system and the system responds to their actions, not the other way around. Users can traverse the news feed, filter the news feed and react to a post, by choice. Users can also just read the posts in the news feed and not interact with the posts, allowing the users a full level of control.

The aim of a professional design for throughout SocialEyes by using the same themes was required so the first golden rule of consistency was followed. The rule states that similar terminology should be used in menus and for navigation. Any 'button' activated with dwell time are all coloured in the same way to make it clear to the user where they can dwell and where they cannot. This was of the main reasons to change the style of the initial landing space. When using Mockplus it was clear that it would be easier to have all the base menus in the same style, so it was easier for the users to navigate the system. The filtering would therefore be placed inside a sub menu, similar to the other commands. In Mockplus pop-up events were added, which showed the system worked well with the small design change. All the radial menus were designed to be the same size and have the same position on the screen. All slices when toggled in the radial menus were turned purple again to keep consistency in the application.

It is also vital in the system to permit easy reversal of actions. In order to change settings in the system, they toggle filters and options on and off. It takes the set dwell time of the system to reverse a choice and is very straight forward to achieve. Whenever a menu opens, the user can exit the menu by using the close menu symbol, to quickly reverse the action they had performed.

Reducing the short-term memory load is also a rule to consider. Shneiderman states that the reduced ability to process information in short-term memory, requires that displays are kept simple, multiple page displays are reduced to single page displays and sufficient training time is allowed for codes, mnemonics and sequences[32]. A white background is used throughout the application in order to keep simple and not clutter the screen distracting the user. The basic graphics used for the scrolling allowed it to be clearly understood by the user. The buttons were clearly labelled, showing their functionality which therefore limited the amount of time a user needed to train on the system to remember what all the functions of the system were. An assumption for the system was that users would be familiar with the labels as it would be similar to familiar social media terminologies.

Another useful feature that comes from the eight golden rules is enabling frequent users to use shortcuts. Traditionally a shortcut would be a keyboard shortcut, but this does not fit with the principle of SocialEyes only having eye tracking input methods. Therefore, SocialEyes would have its own style of shortcuts. One instance of this in the program would be the use of the home button, which would stop the users having to scroll to the top of the feed once

they reach the bottom. This shortcut of going to the top when it would be beneficial for the user comes with the users with more considerable amounts of experience. Another form of shortcut was SocialEye's ability to filter out items that users no longer wished to see, allowing for quicker traversal, which would be used more by the experienced users.

Finally, another golden rule is the ability to offer informative feedback, relating to the size of the action. The most significant action that can be performed by a user is filtering the news feed, with many actions running on the back end of the system. When the news feed is updating the user is shown a loading percentage, which is shown instead of the news feed. This number shows the user how close the program is to give the user the new look news feed and once it is loaded the news feed appears and the loading percentage onscreen disappears. Another way that SocialEyes offers feedback to users is how it shows that menus have been toggled or not, by changing the fill of the slice from white to purple it is apparent to the user what has been turned on and what has been turned off.

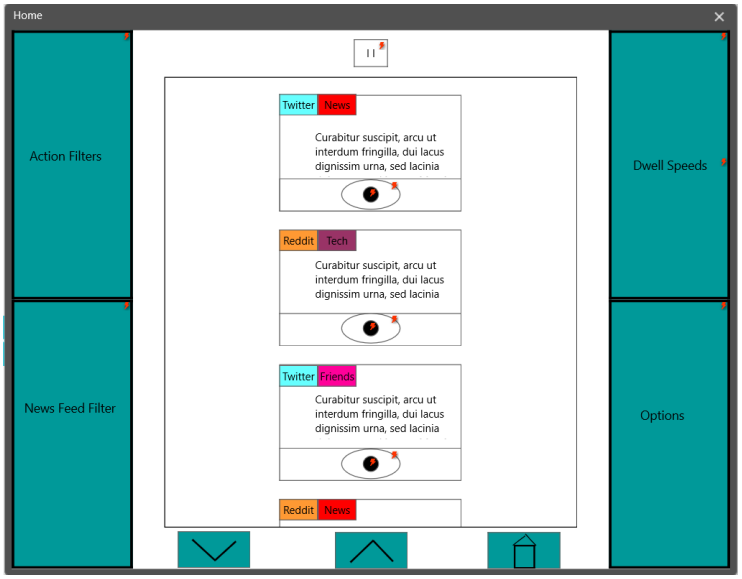


Figure 11: Main area of the application showing central feed and menus.

With Shneiderman's rules used when designing the mock application along with the information gained throughout the design phase of the project a prototype was created (see figure 11).

Care was required when choosing the colours of the buttons. Twitter and Reddit was the chosen social media platforms, both of which have colours associated with their image. Humans associate a colour with different entities so it

was important to use Twitter and Reddit's set colours, which meant the colours of the buttons had to be a colour that could follow the effectiveness principle. Munzner in Visulization Analysis and Design the concept of discriminability where "if you encode data using a particular visual channel, are the differences between items perceptible to the human as intended?"[18] The green colour selected for the buttons was different from the other colour so it would always be apparent to the user what buttons were activated by dwelling.

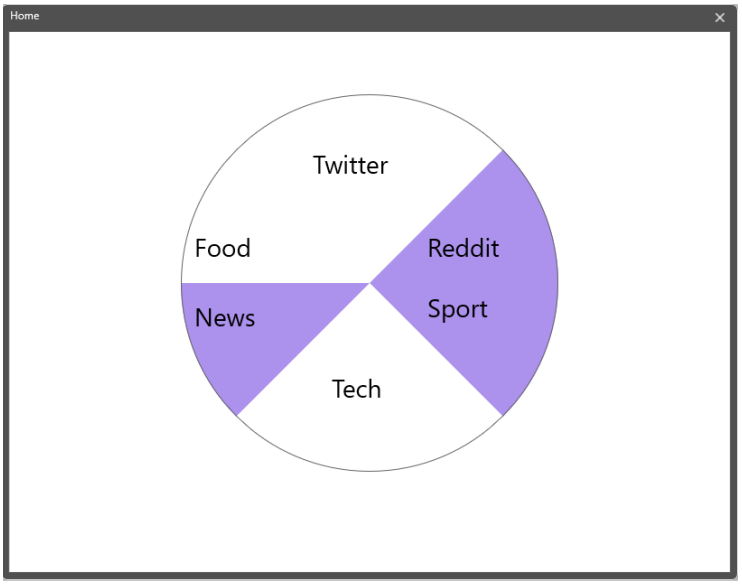


Figure 12: Filter options for news feed with various options toggled.

The menus down the sides of the application had been altered to allow more selections via radial menus, which when dwelled upon brought up a radial menu (see figure 12).

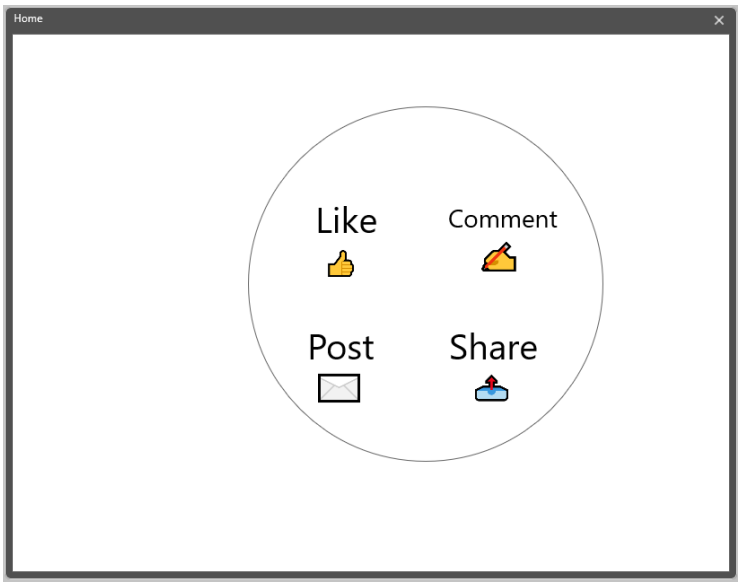


Figure 13: Radial menu to react to a post.

The use of icons on the menus also helped users learn and gain experience with the application. "Icons when used correctly, can enhance usability, be easily remembered and improve the design of the webpages of software being used"[40]. Icons save time when it comes to users reading what the slices action is, which can form an issue due to buttons having dwell times. Icons make it less likely for a user to make a mistake by reading a button. Icons offers more consistency and familiarity, with the icons in the system being similar to icons used in current social media platforms, allowing the user to make an easy connection between them (see figure 13).

The Mockplus application created allowed quick traversal with artificial dwell times being included to allow the researcher to gain a feel for how long to make the dwell times during the implementation.

8.5 Back End Design

With the front end and functionality of the system designed, how the program would be oriented was then investigated. The Tobii EyeX tracker feeds an X and Y coordinate to the program, with no other information, which is reminiscent of several game engines, such as Phaser[19] and Processing[22]. This concept led to the idea of a game loop being investigated to handle the input of the eye tracker. The idea of the game loop is that every frame a series of checks occur to change how the user interface appears to the user, which then triggers different

methods to run. The following list of features would need to check inside the game loop;

- Are the eye tracker coordinates over a selection area with dwell time?
- Is the eye tracker still over the same selection area?
- Has the eye tracker been over the selection area for the required dwell time?
- Can the eye tracker run the command that has been selected?
- Will the command effect the news feed?
- Carries out command selected by the user and updates the UI.

The list of features are demonstrated in figure 14.

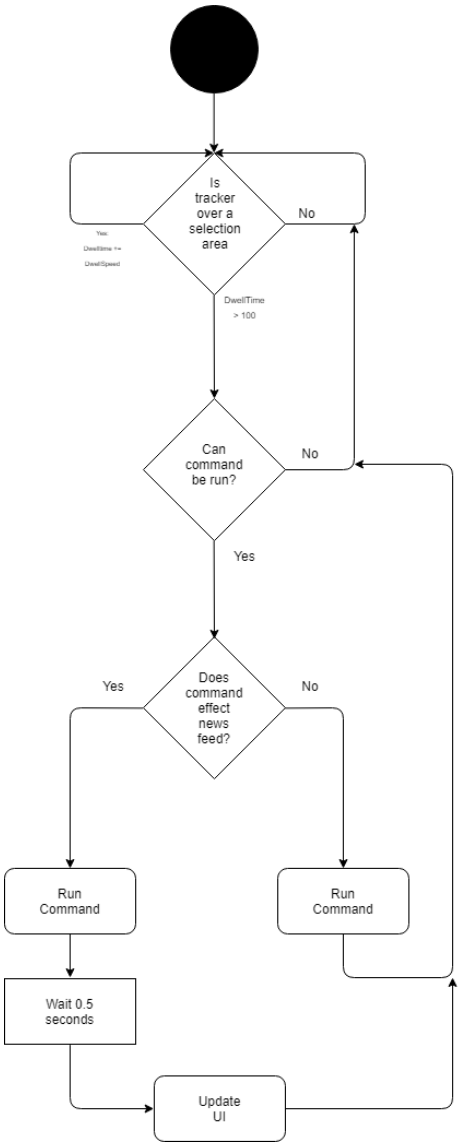


Figure 14: Game loop continually running in the back end of the application.

9 Implementation

9.1 Technologies used

9.1.1 Hardware

Different ways of handling eye tracking were investigated due to it being the only source of input for the system. A simple web camera built into a computer was one option, and Cheung and Peng published a paper describing this idea[13]. In the paper, they show results with gaze tracking and results with head movement.

The accuracy of the gaze movement had a value of 1.28 degrees. The accuracy of the tracker worsens the further from the centre of the screen the eyes position are and with head movement introduced the overall accuracy of the system degrades to 2.27 degrees.

Another option than using a webcam was to use specialised hardware. The Tobii EyeX eye tracker is an infrared-based camera that tracks a user's gaze[33]. The tracker does not track head movement, but considering the applications target user base being fully disabled users, the lack of the future does not affect the hardware choice of the system. The EyeX has an accuracy of 0.5-1 degrees, much better than the webcam. The cost of the EyeX is considerably lower than its alternatives, and with its higher accuracy, it was chosen for the hardware of the project.

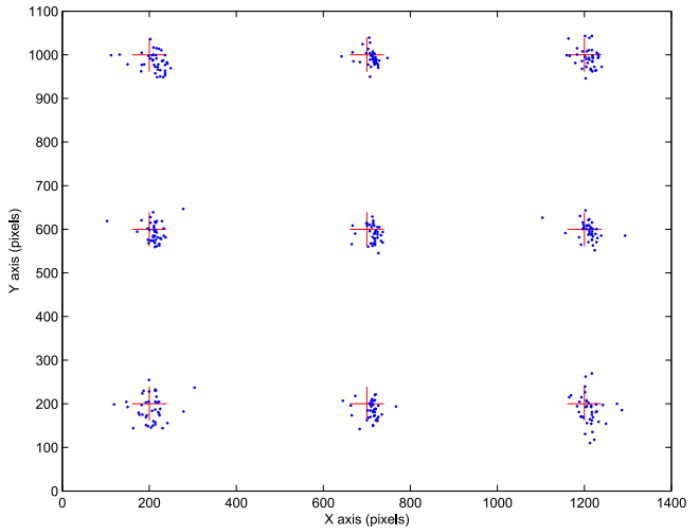


Figure 15: Eye tracking selection areas on a screen.

The different gaze points of the eye tracker have different levels of accuracy.

The more central the gaze of the tracker the better the accuracy of the tracker (see figure 15). In order to counter balance this calibration would be performed when a user initially uses the system. The calibration process would be handled by Tobii's software and involved the user gazing at certain areas of the screen. In total calibration takes under a minute and is only performed when a user first sits down to use the tracker. The lower accuracy around the outside of the screen was mitigated due to the nature of radial menu slices getting larger the further from the centre of the screen they are.

9.1.2 Software

By selecting the EyeX, the programming languages, that could be selected were more limited than using the webcam. Several languages were investigated; JavaScript, Java, Python, and UWP. The EyeX natively supports Python, so was the first choice. A specialised library was created by Tobii to handle all the eye tracking inputs. However, this was not supported by the EyeX. Both Java and JavaScript had libraries available online to stream the gaze data.

Both of these methods were investigated, with basic eye tracking programs implemented. The Java library was run through Processing, which had an external command line program streaming the eye tracker data to the Processing program. While the JavaScript program involved transferring data using Node.js. However, the issue with both of these methods was that the data could occasionally lag, which would affect the speed and accuracy of the software.

Finally, Universal Windows Platform (UWP), which is based on C# and XAML was investigated. The platform supported the core Tobii gaze data with no lag and also comes with an easy to use design area for the UI section of the implementation. The language chosen was UWP due to the smooth gaze data and the additional support offered when created the UI.

9.1.3 Operating Systems

The operating system selected for the project was Windows 10, due to the built-in eye tracking aids that are supported by Windows when using a Tobii eye tracker.

Figure 16: Microsoft assistance bar.

The menu includes methods for different mouse clicks, selecting a particular area of the screen and also a build in keyboard[15] (see figure 16). With entering text using eye tracking being its own research area, having the built-in keyboard would allow for more time to be focused on SocialEyes.

9.2 Handling Eye Tracking

The eye tracking input to the system needed to be handled, so a method was created, in which the program would first check if any devices were connected and attached to the system. If a device was found the system prompted the user to grant access between the program and device. The UWP API was used to gain the x and y coordinates of the user's gaze. A built-in gaze preview object from the UWP API was used. This has checks that include whether the tracker can track eyes and whether the tracker can be successfully configured. After this a gazedevice watcher from the API was set up that contained an update function which performs some final checks. This then calls the GazeMoved() function, which translates the eye position into x and y coordinates, which can then be used to track what users are looking at.

The eye tracker's position is decomposed into a x and y value with the method DoesElementContainPoint() then called. The initial part of this method involved calculating the angle that the eye is looking at compared to the centre point of the screen. A filter is applied to the gaze point coordinates in order to have their angle compared from a (0,0) centre instead of the resolution of the screen. The radius is calculated in order for the user to be able to select any slices on any radial menus, with the values calculated continually in order to check if the gaze is inside a selection target.

The next part of the method was a series of checks to work out if the current dwellable location contained the current gaze of the user. These checks involve checking the coordinates of the gaze along with the radius and angle of the gaze, if a check was valid, then the current segment was recorded, and the method continues. At the end of the method, the current dwell time on the area was calculated. If in the previous check of the gaze the current segment was different or nothing then the dwell time counter was started from zero, else it continued. Once the dwell time counter had ticked a check again was used to see if the required dwell time had been reached. If so the selection was made.

9.3 Social Media Data

9.3.1 Testing Possible Social Media Channels

In order to populate the central feed, feeds had to be obtained from multiple social media sources. There was a significant amount of time spent working out the best social media to use. The first choice investigated was Facebook due to its massive user base. However, many issues arose when trying to implement Facebook into SocialEyes. In recent years due to widespread data leakage such as the Cambridge Analytical scandal[3] the process of gaining access to Facebook data for use in an external application has become a much stricter process. Developers must apply for approval, giving full justification, but this is a common practise when requesting approval from social media sites. Facebook also now require a video of the application in use, highlighting where the

Facebook data would appear. In a small time scale project such as SocialEyes, this is an issue. It would become high risk if Facebook was relied upon, as after the application development had finished, permission could be rejected. There then would be no time to develop an alternative, due to the time scale of this project. Other issues surrounding the use of Facebook also stem from its new restricted process, such as needing more approval to extract data from pages that are not owned by the user creating the app. Online tutorials suggested that information can be obtained from personal pages, but this had also been restricted[27]. Another issue is that Facebook also had stopped some of the commands that allowed users to share and post to their site. Users have to post using embedded Facebook buttons that have no eye tracking capabilities, which highlights the necessity for this research project. Overall the lengthy process and restrictive nature of Facebook meant it was not fit for the project. But could be considered for future development.

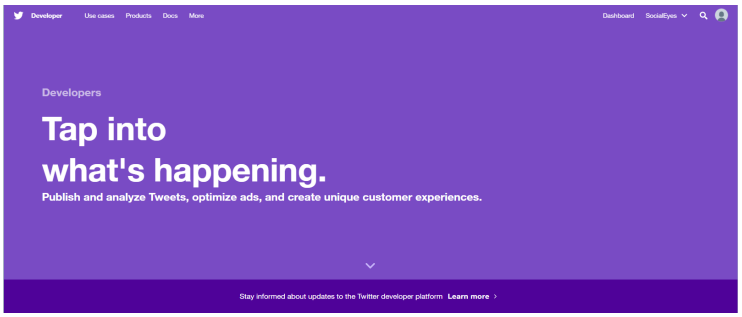


Figure 17: Twitter developer platform used during the project.

The next social media site assessed was Twitter. Similar to Facebook a research account was created for use in the project in order to stop any personal data of the researcher entering the project. With this mock account an app was created on the Twitter Developer Platform[36].

In order to create an app on the platform, a description of the app and a callback URL were required. After the information was passed, a secret key was produced that allowed access to Twitter's data services.

In order to access the data, tools such as CURL or HTTP requests were options. However, this increased the complexity and the likelihood of mistakes. Therefore a popular library called Tweetinvi which is an intuitive .NET C library to access the Twitter REST API was used[35].

The next social media site investigated was Instagram, but as a part of Facebook, it suffered from the same issues, as well as lacking some of the required social media features such as sharing. This led to more open sourced developer

platforms being investigated, which produced the idea of using Reddit as one of the social media sites. Reddit has fewer restrictions in place surrounding the development of an app, with posting to the site monitored by Karma[24]. Users are not allowed to post on feeds if their Karma is low and requests to their API are reduced depending on the Karma of the user. Therefore a section of time was committed to increasing the Karma of a dummy research account in order to have no restrictions placed on API requests for the project.

The same process was applied as with Twitter, where a secret key was gained, but this time two more access codes were required through OAuth token generators. A 3rd party website generated the tokens for the application which were then all the authorisation details that were required to access data from Reddit[11]. After previously using the Tweetinvi library, it was decided to find a similar library for Reddit. A NuGet package was used called Reddit.NET[25], which unlike Tweetinvi was a new, little used library. However, the package was lightweight and could obtain all the information that was required.

9.3.2 Retrieving Twitter Data

In the code of the project to retrieve twitter data, the SocialEyes account was attached to the program with the user's profile, which then allowed a timeline to be retrieved containing the tweets a user would normally see on a Twitter native application.

```
var user = Tweetinvi.User.GetAuthenticatedUser();
timelineTweets = Timeline.GetHomeTimeline(40);
```

Figure 18: Code for retrieving tweets from a user's Twitter timeline, using Tweetinvi.

The variable passed in was the number of tweets retrieved from the top of the timeline. Forty posts were chosen as by the time the user looks at all the posts from Twitter and the same amount from Reddit, they would need to refresh, and a new set of posts would appear (see figure 18).

There is a limit to the number of posts which are retrievable in a period of time, this can be bypassed with a subscription fee, so in future releases of SocialEyes this would not be an issue.

9.3.3 Retrieving Reddit Data

Figure 19: Code for attaching a Reddit account to the program, using Reddit.NET

Retrieving data for Reddit followed a similar process to Twitter, with an account associated with SocialEyes by attaching the keys via a Reddit object. This then allowed the posts to be retrieved (see figure 19).

Once the account was attached, the posts from all the subreddits that a user was subscribed to needed to be retrieved and this could of be done in several ways. From getting the top post, best posts, most controversial and most downvoted post. It was decided to use the GetBest() method from Reddit.NET to retrieve the posts as it retrieved the posts the user would most likely see on the main site.

9.4 Central News Feed

All of the posts from Twitter and Reddit needed to be fed into a central news feed so the user did not need to switch back and forth between two different applications. An element that allowed items to be dynamically added into was required, which could also scroll. There were various ways to accomplish this using XAML and UWP classes, which were Scroll view, Grid view and List view classes. In the ScrollViewer class, the scroll bar is down the side, but no function can be called to make it scroll, which was an issue for scrolling using the eye tracker. However, the ListView class in UWP contains a function called ScrollIntoView(), which scrolls the list view to the position of the item with the index given to the function, while also having a scroll bar down the right hand side which indicated how far down the user was. The ScrollIntoView() function was used to implement the scroll buttons of the program. An index stores the current position of the scroll in the listview. This was incremented or decremented whether the user was scrolling up or down. The home button works by going to the item at index 0 and immediately returns the user to the top of the feed.

When creating a ListView, an ItemTemplate is an option if there are going to be many similar items such as posts from social media sites. Inside the template, the style of how each post would be displayed was created, with the styling coming from the Mockplus application.

Figure 20: A single post as it appears in the central feed.

Two rectangle objects were placed in the top right of another rectangle which contained all of the post information. Three text blocks were created to contain the text of the post, the social media site it was from and the type of post it was. In order to keep all the objects correctly arranged, a containing grid system was used placing the objects into columns and rows. A border was placed around the box so that users could see what text and tags were associated with what post. An example of a post is shown in figure 20.

The challenge that came was feeding the information that was retrieved from the original posts into the central news feed posts. In UWP values from classes can be bound to UI elements. Therefore a custom class was created for a post, whose values were from the post. The class contained a value for the text of the post, social media site it is from, the type of post it was, as well as font sizes for the text and an id to identify the posts later in the program. Values that were returned from functions inside the class, could also be bound. This method allows posts to be customised and dynamically added, such as when the feed was refreshed.

The main use of the dynamic binding is too set the colour of the tags depending on the type of the post. Depending on the type of the post a different value was put into the UWP SolidColorBrush object, with the parameters being converting from hex values into RGB values as shown in figure 21.

```
public static SolidColorBrush socialMediaTextColour(string socialMediaType)
{
    if (socialMediaType == "Twitter")
    {
        return new SolidColorBrush(Color.FromArgb(255, Convert.ToByte("#ffffff".Substring(1, 2), 16),
        Convert.ToByte("#ffffff".Substring(3, 2), 16),
        Convert.ToByte("#ffffff".Substring(5, 2), 16)));
    }
    else if (socialMediaType == "Reddit")
    {
        return new SolidColorBrush(Color.FromArgb(255, Convert.ToByte("#000000".Substring(1, 2), 16),
       Convert.ToByte("#000000".Substring(3, 2), 16),
       Convert.ToByte("#000000".Substring(5, 2), 16)));


    }
```

Figure 21: socialMediaTextColor function to get tag colours.

A similar piece of code was also run to change the colour of the text that appears on the tag as using purely black text meant some labels were unclear to the users (see figure 21). The text from the post font size changed depending on how much text was inside the post, with there being three ways it could be displayed, large font, small font, or shortened text with a full-text version then available through the react menu. The object was added to the ListView once all values had been calculated.

9.5 Radial Menu Creation

After the radial menu design was chosen as the central menu for the system, a method for creating one had to be established. In UWP the graphical side of the program has a workspace where items from their toolbox can be placed. The initial idea was to use the ellipse tool along with line tools to segment the ellipse. This method would form a lightweight solution which could be quickly implemented to produce a visual element that was easily adaptable to different numbers of menu items. However, detecting which segment of the menu was being gazed upon was not so simple. Therefore, event handling would need to be created to handle the inputs. The main issue was being able to alter how many options would be present in a menu and performing this with a line tool would cause inaccuracies and a greater load on the processor, therefore it was not suitable. In UWP NuGet is used as the packet manager for .NET libraries.

The RadialMenuControl package was used for the rest of the implementation of the project. As with any UWP application, most UI elements can be created with either XAML or with C#. For this project as there was a Mockplus application produced during the design phase it was reproduced on the XAML side. With the radial menu in the toolbox, it was placed into the workspace which then auto-generated the code that represented the menu. Within the code, the diameter of the menu changed, and the start angle was set to 0 so that the items in the menu would be oriented from North. To increase the number of slices contained within the menu another RadialMenuButton component was added, with each of the components having a name, label and icon. Different icons appear throughout the radial menus in order to gain the full benefit mentioned

in the design section[8].

9.6 Filtering

As well as being able to traverse the feed with the scroll functionality, it was important for users to be able to filter the news feed. As the central feed contains posts from multiple sources, it was important for the user to be able only to see one of the sources if they wished, so the filter would contain all social media sources. Also, the user should be able to filter out different types of posts. Therefore all tag types were also included in the filter.

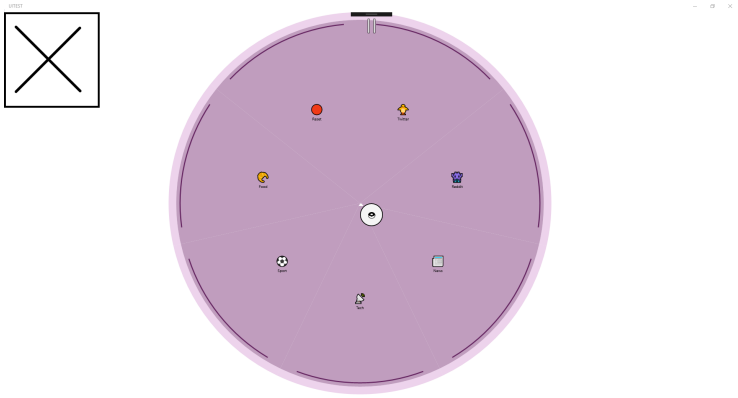


Figure 22: No filters applied to the central feed.

The quick, easy reversal of actions discussed within Schneiderman's rules is also shown within the filter, with there being a reset section to the menu which turned all filters back to being off (see figure 22), fully populating the news feed again. Similar to the actions filter menu of the system where the menu was highlighted to indicate whether they were on or off (see figure 28).

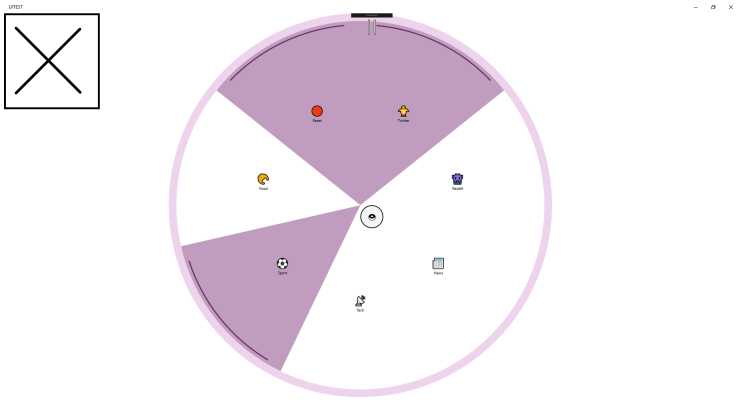


Figure 23: Partial filters applied to the central feed.

The items were filtered out by clearing the listview then repopulating it, not including any items tagged in the same category as the applied filter. Two different methods were developed which can assign tags to posts; using common words, and using Machine Learning.

9.6.1 Tags

The first and more straightforward method investigated was using common words and the information from a post to decide how to categorise it. Nearly all social media platforms have a description section for a page that a user is following, using this along with the name of the account can be used to help categorise the post. The text within a post was another option, but not all the words would prove useful. Hence Natural Language Processing needed to be performed.

The data from the posts were preprocessed, removing any unwanted characters and converting all text to lower class. Each of the different sections of text checked against a range of words closely associated with the topics. If they contained one of the words the assignTag functions returned the category it should be placed into, which in turn was passed through to the news feed assignment section. The words used to check against were sourced by scraping the descriptions and names of the top pages and subreddits in the topics on their respective sites.

The assigning tag method proved very quick and accurate, with only a few recorded examples of mislabelled posts. This was due to keywords appearing in multiple categories such as a nutrition post about an athlete, placed under sport and food. Therefore a more advanced solution was investigated.

9.6.2 Machine Learning

Using the naive tag assignment relies heavily on knowing where the post has come from as well as having a description of who has created the post. However, there also needed to be a solution if a blank post with no description was found, for example, a post made by a friend or a new user.

The field of Natural Language Processing (NLP) along with Machine Learning (ML) was used to classify social media posts into categories. Different machine learning methods were implemented, with a comparison score calculated to decide the best method for the problem.

The data that was used for training was obtained online[34] and was cleaned before it was used by removing any blanks and corrupted entries. The data files were converted to csv files and then combined into one file which was given column headers in order for the program to access the data. The combined file contained a column with the text of the post and another column with the category of the post.

For all the models created the data was read in, with Unicode characters accounted for. A list was created for both the descriptive data and the categorical data. The sklearn library was used to vectorise the lists, as machine learning classifiers need numerical data passed into them. A sklearn function called TfidfTransformer() was also used. Tf means term-frequency, which is a common weighting scheme in information retrieval. Instead of raw frequencies of occurrence of a token in a given corpus the goal is to scale down the impact of the tokens that occur frequently, which hence are less empirically informative than features that occur in small fractions in the corpus.

The data was then split into testing and training data, to test the classifiers that were chosen.

Three different methods of classification were chosen. These were Naive Bayes, Support Vector Machine (SVM) with a linear kernel and a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel.

A Naive Bayes classifier is straight forward classifier, which is used for NLP problems. The classifier is based on Bayes Theorem, which works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. Using the conditional probability, we can calculate the probability of an event using its prior knowledge.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

A SVM is a supervised learning classifier, which uses hyperplanes to create zones where items of one category will likely appear in. The different kernels, allow different styles of hyperplanes.

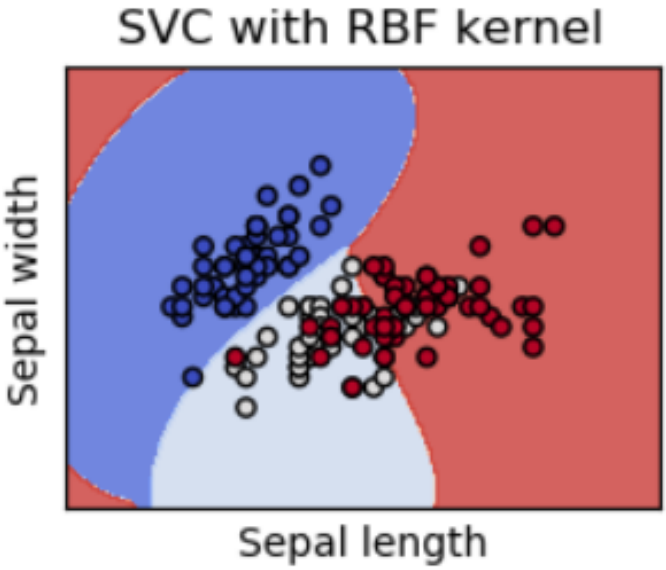


Figure 24: Support Vector Machine using RBF kernel

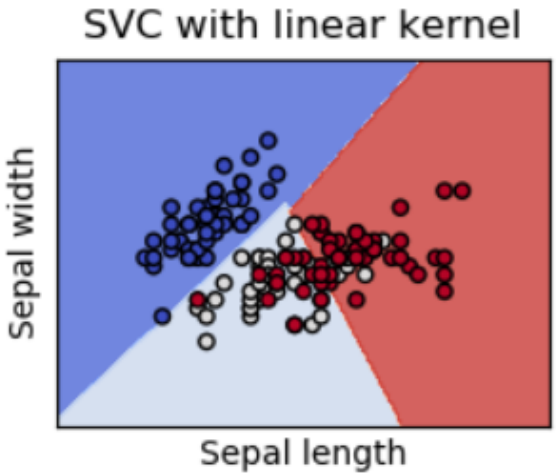


Figure 25: Support Vector Machine using linear kernel.

In figure 24 and figure 25 the different types of hyperplane can be seen. The type of kernel that worked best was calculated through running a score calculation on the classifiers.

Classifier	Accuracy
Naive Bayes	76.58
Support Vector Machine (linear)	96.21
Support Vector Machine (RBF)	40.49

The table above shows that the SVM with a linear kernel classifies social media posts at an accuracy of 96 percent. A SVM is a more complex classifier with a longer training time, but as the results were better it was chosen. The linear kernel was also chosen due to its high performance, which was because linear kernals perform better than RBF over large amounts of data such as the combined data used for SocialEyes.

The machine learning offered high category classification accuracy and was set up to take in the text of a social media post. Due to time constraints it was not feasible to pipe the results into the C# code from Python.

9.7 Action Filtering

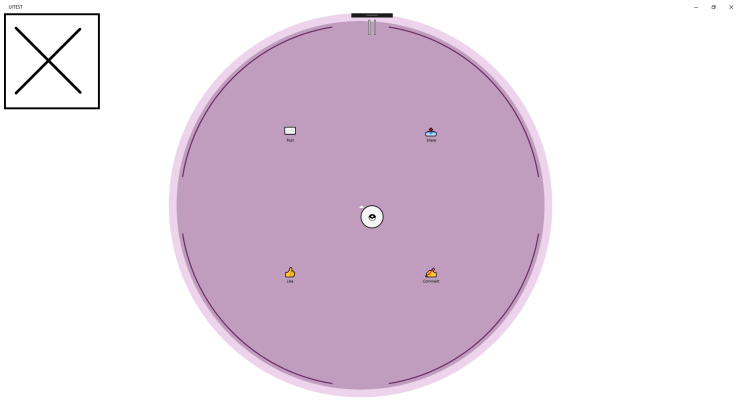


Figure 26: Showing the options selected when filtering social media features.

A core feature of SocialEyes was the ability to customise the actions that a user could perform. The idea being that not all users want to use all the features offered by the social media sites, therefore the input menus should not contain them (see figure 26). By only having desired features the input area of each section of the radial menu increased. For example, two items with have half the menu each, instead of a quarter each with the rest of the menu being wasted space. It had to be possible for the user to easily filter and change them so the application could always fit the user's needs.

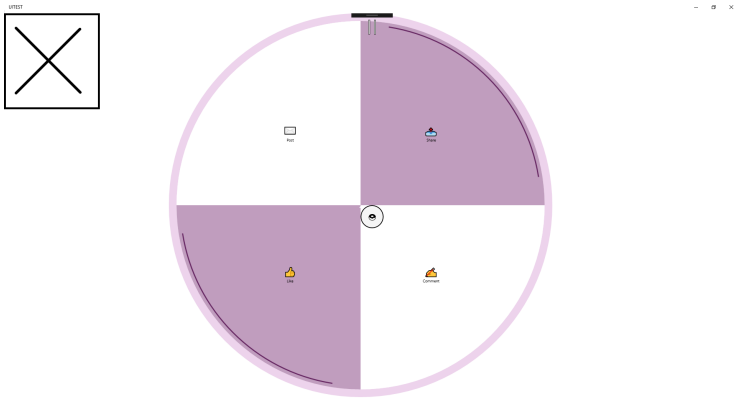


Figure 27: Showing only two options selected when filtering social media features.

As stated in section 8 feedback would be given to the user, and part of this was showing what had been selected already by the user. The radial menu contained a command to toggle individual sections of the menu if the user dwelled

on a section of the menu for the current dwell period. An example of a toggled menu is shown in figure 27.

There were four radial menus created in the XAML code for the different features that the user could use within the application. When the program is first run the number of features in use decides the menu that will appear. An array containing the features being used was looped through so that the radial menu can have the correct labelling and icons. There was a downside to the use of the radial menu when compared to the custom alternative, which was activating the hovering event which highlights the option the user is selecting. The event was not possible to call with the eye tracker, but due to the low dwell time, this was not deemed to be a significant issue.

9.8 Changing the Settings

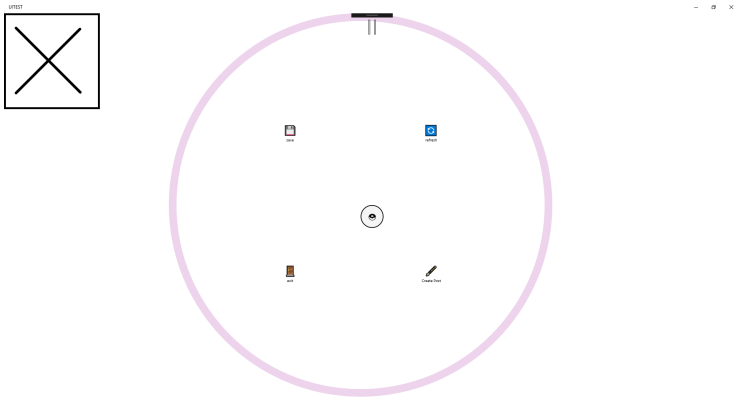


Figure 28: Options list radial menu in SocialEyes.

In the bottom right of the main screen, there is a large dwell area for the user to activate, which launches a radial menu (see figure 28) with the following options;

- Save
- Refresh
- Create Post
- Exit

The custom nature of the system loses impact if the users have to set the application to their specifications every time the application was opened which would cause a significant time loss. Therefore several ways were investigated for

how to best implement a saving system.

The first one was the use of a database to store the settings and keys for each user of the application. Databases are suitable for storing large scale data stores, while also protecting the keys if the database is correctly implemented with SQL injection protection and encryption. The scale of the database for the amount of data needing to be stored was too high; therefore an alternative safe system was investigated. Programs designed for disabled users will typically only have one user log into it on a device, therefore storing keys and having an account management system was beyond the scope of the project. The information to be saved was the social media features in use, the dwell speed and the filters applied to the news feed.

In the end, a simple file that contained the required information was written to when the save segment of the radial menu was selected. When the program first starts up this information will be read and the program will cycle through the data and set all values to those read from the files.

The refresh segment of the menu when dwelled upon will populate the central feed again by calling the same methods run when the news feed is initially created.

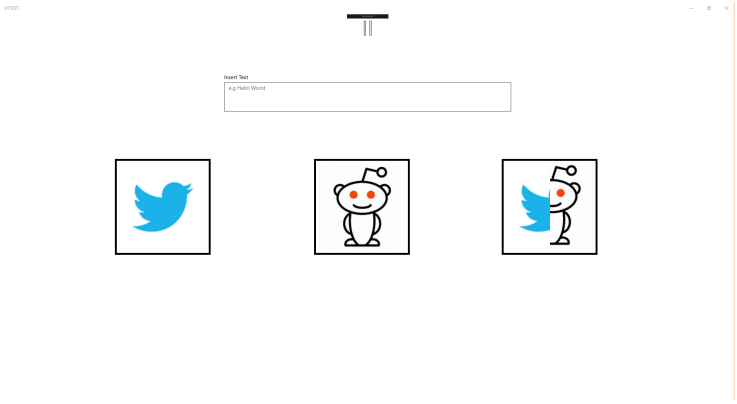


Figure 29: Area of SocialEyes for creating a post.

The *create post* section of the program (see figure 29) works as a hybrid, being a cross between the *share menu* and the *comment menu*. The text for the post comes from the text block at the top of the screen, with the user then dwelling on the site/sites they wish you post too.

Posting to the platforms works on both sites by creating a post object in the respected libraries, then calling post methods, passing the post objects into

them, which resulted in the post appearing on the correct websites.

Finally, the exit section of the radial menu when dwelled upon ran the CoreApplication.Exit() function which would immediately exit the program.

9.9 Dwell Times

The amount the counter increased by was dependant on the dwell time that had been set by the user. The program had three speeds to choose which were changeable in the dwell time menu at the right of the central area. Once the dwell time was reached, various functions were callable.

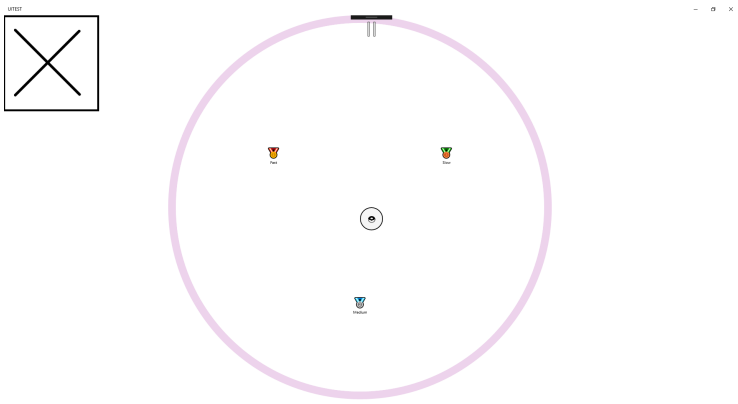


Figure 30: Radial menu to change dwell speed of SocialEyes.

As stated one of the functions is when the user has been staring at the dwell options. The menu is straight forward with icons added to the menu to help demonstrate to the user the range from slowest to quickest dwell speed, which can be seen in figure 30. The slow dwell time was 3.3 seconds, the medium dwell time was 1.1 seconds and the fast dwell time was 0.6 seconds.

9.10 Reacting to Posts with Common Social Media Features

As discussed the user might not want to perform every one of the social media features and the different features can be toggled on and off as with the previous filter menus.

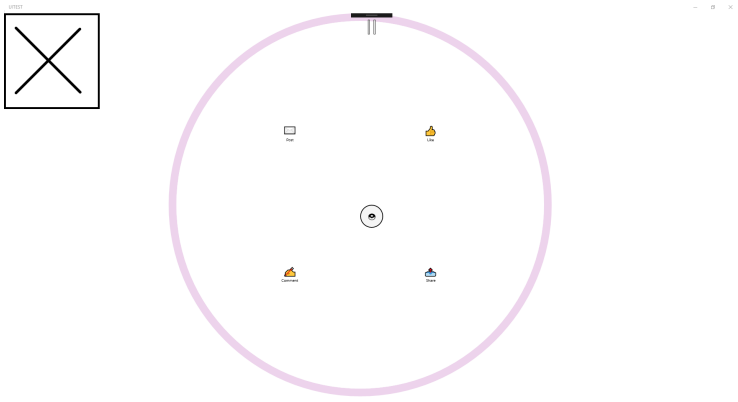


Figure 31: React menu with full set of social media common features.

However, when the options are toggled off, they have a much higher space allowed to the users.

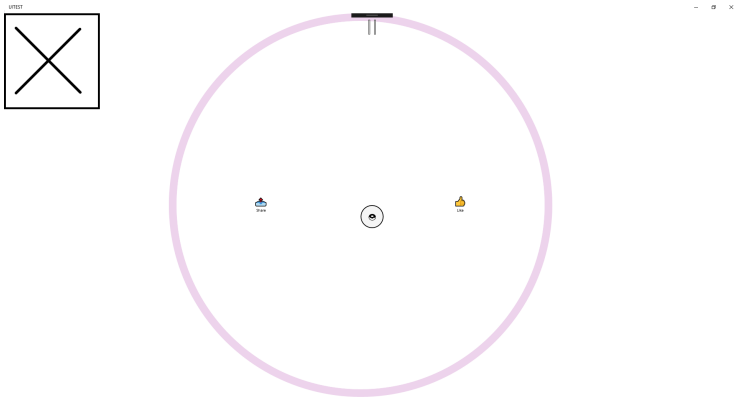


Figure 32: React menu with partial set of social media common features.

The functionality of the program that allows this was a list of the functions that were currently in use from the filter menu. The size of the list was used to decide which radial menu to bring up on the screen. In figure 31 and 32 we see radial menus for four features and two features respectively. Once the menu had been selected, the icons, text and functions to be called needed to allocated to the slices on the menu. This was achieved by iterating through the list of functions and assigning them to slices, by setting the *wedgefunction* variables to the name of the feature.

The text blocks on the screen alter the XAML code, and the icon variable attached to the radial menu were altered. The wedgefunction variables were

used to call the correct functions. The dwelled upon area was then passed into the beginAction() method, which individually called different features.

When a user reacted to a post, it was essential to know which post had been reacted too as well as which social media site the post is from, so that the correct code rans on the correct post. Whenever a post was added to the ListView the button created in the XAML either contains Reddit or Twitter in its name, denoting the social media site the post was from. The id of the post was also attached to a post in the news feed which made it findable in the list of posts.

Once the post is found it is passed to the individual methods that were used for the different common social media features.

The *like* function was the most straight forward of the features, with both Twitter and Reddit not requiring any other input than dwelling on the Like segment. In order to like via the Twitter API, the command $Tweet.FavoriteTweet(tweetToUse)$ was run and for Reddit, the command $item.Upvote()$ was run. These can be undone by the user calling the like function again, which then runs an unfavourite and no vote functions in each site respectively.

Figure 33: Options presented to the user when sharing a post.

The share function was more complicated compared to the other features. After the user has selected to share a post another set of menus appear.

In this menu section (see figure 33), the user chose which sites to share too. To make a selection the user dwelled on the image that they wished to share too. Through this menu, it was possible to share from Twitter to Reddit and vice versa. It was also possible to share to both platforms at the same time.

```
else if (option == "shareToBoth")
{
    if (currentMedium == "Twitter")
    {
        var tweetPost2 = Tweet.PublishRetweet(tweetToUse);

        var sub = reddit.Subreddit("r/SocialEyesPosts");
        SelfPost myPost1 = sub.SelfPost("Post", tweetToUse.FullText).Submit();
    }
    else if (currentMedium == "Reddit")
    {
        var tweetPost3 = Tweet.PublishTweet((String)redditPostToUse.Listing.SelfText);
        SelfPost myPost2 = (SelfPost)redditPostToUse;
        myPost2.Subreddit = "r/SocialEyesPosts";
        myPost2.Submit();
    }
```

Figure 34: Code showing how a Twitter post was shared to both platforms.

In order to be able to do this the text of the post needed to be stored (see figure 34) and was used later when creating a new post to share to a new platform. After a post had been shared the menu closed and returned to the main area.

The comment function works the same as the previous functions with another menu system brought up when dwelled upon by the user.

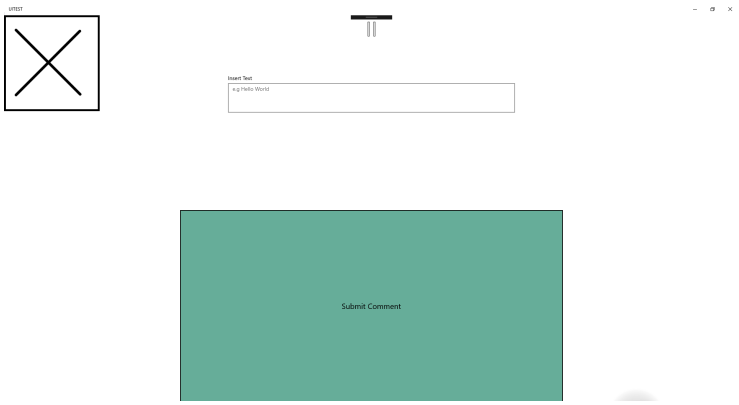


Figure 35: The UI for commenting on a post.

For this menu in figure 35, there was a text box for the users to type into, with the user using the Microsoft assistance tool to type in the box. Once the user had finished typing the message, they dwelled on the large submit comment area, which submitted the comment on the box unless the text box is empty. Both share and comment returned the user to the main area.

Figure 36: Example of a post where entire text is not visible.

The final option inside the react menu focuses on allowing the users to see the entire text of a post. This was required as text maybe truncated in the main feed (see figure 36). This option was placed inside the same menu as the other features as users would likely want to see all of a post before they reacted to it.

Figure 37: Example of a full-screen post.

Once a user had finished reading the post shown in figure 37, they used the cross in the top left of the screen. By dwelling on this area, a function created called closeOpenMenu() ran which closed all the open menus on the screen, returning the user to the main area of SocialEyes.

9.11 Handling the Updating of the UI

A major problem that was encountered during the development phase of the project was related to the XAML UI not updating quick enough when compared to the code running on the C side. On several occasions, reactions to posts and filtering of the menus did not appear correctly. This was difficult to diagnose, as while debugging the issue did not manifest. This was because the UI had the

time to update correctly in debug mode but not in deployment mode. This was discovered by converting any methods that needed to wait into asynchronous methods, which meant the program waited a period, for some data to arrive before executing the next line. By including these small waits into the code both the UI and the back end ran with all the correct data in the right place.

10 User Study

10.1 Design

The object of the study was to compare SocialEyes and the current alternative, which is running browser versions of the social media accounts. The input for the browser side of the study was handled using the eye tracking alongside Microsoft's built-in eye tracking controller[15].

All the features implemented in SocialEyes would appear in the user study. The five features selected would form five tasks that each user would perform.

- Hypothesis - 1.H_0A: There will be no difference in speed between SocialEyes and the Browser.
  2.H_0B: There is no difference in accuracy for different dwell times.
  3.H_0C: There will be no difference in accuracy between SocialEyes and the Browser.

- Independent Variables - input method: three dwell times of SocialEyes and system used (SocialEyes or Browser).

- Dependent Variables - Period for a task and accuracy of completing the task.

- Participants - All participants in the study are students from the University of St Andrews. In total ten took part in the within-subject study. One of the features of the system created was its ability to change the amount of dwell time needed to select an item. Thus making it four runs through the five different tasks, meaning each user performed a total of 20 tasks.

- Apparatus - A Tobii EyeX tracker, a wide screen monitor, PC running Windows 10 and Mozilla Firefox. Timing was measured in seconds. The timing for SocialEyes was managed using the Stopwatch class in C#. The accuracy was calculated by a counter, which is iterated every time a selection is made. The minimum number of selections required for the task was divided by the counter which is then multiplied by a hundred to get a percentage. The timing for the Browser was handled by the network tab of the developer tools inside Mozilla Firefox. A .HAR file was then

downloaded and analysed using an online tool[6]. Each task when completed had a specific command name and a timestamp. The difference between command's timestamps was the time. The accuracy was calculated in a similar method to SocialEyes, with all posts clicked, turned into a percentage using the minimal number of clicks it takes for the task.

- Study Procedure and Set Up - Each participant completed the study in the same controlled environment, with researcher supervision throughout the study. Every participant was given a participant information sheet explaining the purpose of the study, also getting time to ask any questions. After they signed a consent form, both the systems were demonstrated, showing how to traverse them. Afterwards, the users were given five minutes to practise, performing some practise tasks. The purpose of this was to eliminate the possibility of the first couple of selections being slower due to the user learning the system. Before any participant used the eye tracker, it was calibrated, in order for the tracker to be accurate. They were also informed to not move around in the study as this could affect the calibration of the eye tracker. Any users who wore glasses had to remove them as they stop the eye tracking working correctly. Any dwell selection other than scrolling would be a selection for SocialEyes, and any click on a clickable section of the Browser would be a click. The reason scrolling was not counted as it could not be controlled within the Browser. Timing begins for SocialEyes when the central feed finishes loading, and the browser begins when the network tool is refreshed. The timing ends for SocialEyes when the user selected the exit area and the end timing for the browser was the timestamp of the final command, which was liking a final post.

The choice of whether to make the study a between-study or within-study was the main design decision that was required. In a between-study, a user would only use one of the applications[1], while in the within-study users use both applications. The concept of a between study is to avoid skill transfer effects. However, the variance is not then controlled within the participant, meaning the participants are a source of error. In order to counteract that, more participants are required. Due to the timescale of the project and the number of participants required a between study would not be feasible to run.

In a within-study, the participant uses all forms of the applications involved meaning that variance is controlled within the participant. Therefore fewer participants need to take part. Skill transfer is the major disadvantage and needs to be reduced with counter-balancing measures.

The order of the tasks completely changed in order to counter-balance any possible learning effects. This was done using a Latin square, which is used when calculating completely accurate counter-balancing is impractical due to a

combinatorial explosion. However, this only reduces but does not eliminate the learning effect. Due to the nature of the project, the majority of users will not be used to eye tracking input. Therefore users were granted time to get used to the system. Each user was allotted the same amount of time to get used to both of the systems, in order to not grant any form of bias to either system.

Tasks issued during the user study;

Task1 The user had to find a post from Twitter, commenting on it with the phrase 'I love this.

Task2 The user had to share a post from Reddit about Technology to both platforms.

Task3 The user had to like a post from Reddit and a post from Twitter in either order.

Task4 The user had to find a post about food, stating the food mentioned on the post, then ending the program/liking the post.

Task5 The user had to create a post stating 'SocialEyes tracks my eyes' and post it to Twitter.

As mentioned in the ethics section of the report a participation sheet was given to the user. Finally, at the end of the study, the participant was given a questionnaire. Within it they were asked to answer the following questions;

- How easy the user interfaces were to learn?
- How easy it was to like, comment and share items from the news feeds on the respective systems?
- What could be improved about the SocialEyes application?
- Any general comments surrounding the SocialEyes application?

The questionnaire was given to the participant to answer without the researcher's input. If a participant was confused, they could ask for clarification from the researcher, but care was taken to not bias or lead the participant.

10.2 Initial Run of Study

Issues arose after two people had taken part in the study. The initial expectation for the study were that due to the nature of an unusual input to the system, the user would learn the layout and how to input to the system through the slow and medium dwell settings, before moving onto the fast setting. However, users found that the slow and medium dwell time's periods were too long, and the flickering that can occur from a user's eye caused the gaze aid to make the user's feel unwell. They both stated they were straining their eyes by staring

at a single point of the screen for those periods. Also, they found that the fast was easy to use and had no issues in becoming proficient in it.

After receiving this feedback it was clear that the study design needed revision, the first two participants took part in what became the *pilot* study. Their data does not appear in the final statistical analysis, as the study they took part in was different in setup to the final study carried out. The length of the study also exceeded the expected time. In the pilot study the length was around one hour, but a length of half an hour was the desired time.

10.3 Redesign of Study

The pilot study showed that the point of interest of the study was towards the higher speed end of the scale. The slow and medium speeds were deemed to be obsolete with two other speeds used instead. The original fast speed became slow, with two new dwell speeds introduced for medium and fast. The dwell times of these speeds were 0.43 seconds and 0.33 seconds respectively. By also having fast as the slowest dwell time, the user's would not have to stare at the screen in a single area for an extended period, which should remove the feelings of sore eyes and sickness from the users in the pilot study.

Finally, one of the tasks was also removed to reduce the time of the study. The difference in posting and commenting on items was minimal, with the same number of user inputs required for each on both platforms. Therefore the *post item* task was removed, which meant the study still had the same number of speeds in the created system and current system and there was one less task for the participants to complete. Therefore the redesigned study would contain Tasks 1 - 4.

10.4 Experimental Results

If there is a difference in the measurements that are obtained via the dependent variables when independent variables have been manipulated, then the differences can be due to manipulation of independent variables or due to an error, where what was measured was down to random chance. Therefore significance tests are run to decide whether the measured differences are statistically significant or not. However, if it is possible to reject the null hypothesis H_0 then we can state we have a significant result at the predetermined confidence level. For this study the confidence level selected was 0.05.

In order to calculate the significance of the statistics gained from the experiment, IBM's SPSS Statistics Data Editor was used[7]. The measurements were compared for each of the tasks checking if the null hypothesis could be rejected or not. First, data was inputted into the dataset section of the SPSS editor, with each of the speeds/methods becoming the variables. Next, a general linear model performing a repeated measures ANOVA was created.

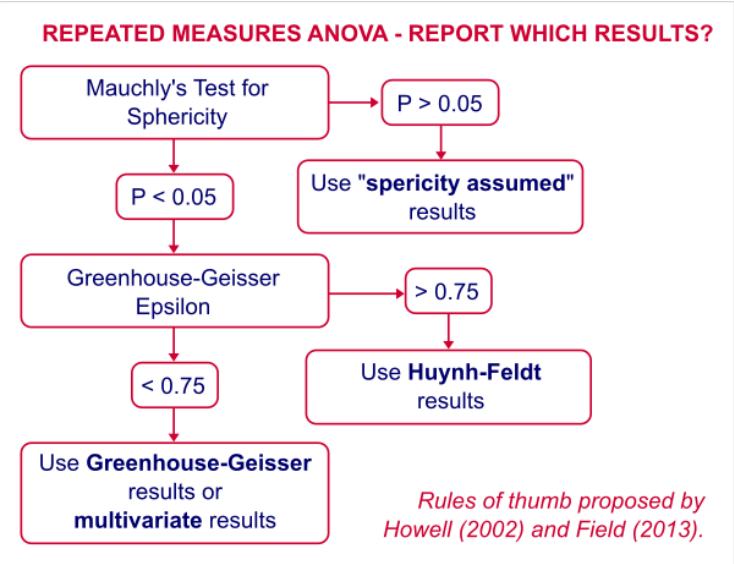


Figure 38: Which results to report for ANOVA[26].

For each of the tasks, the diagram in figure 38 was followed in order to determine whether a significant result had occurred. The value for Sphericity would be required which is the condition where the variances of the differences between all combinations of related groups are equal. To calculate it the Mauchly's Test for Sphericity was checked to see if the significance value was above 0.05. If this was true the *sphericity assumed value* was checked to see if it was below 0.05 and if so the result would be returned as significant. If the Mauchly's Test for Sphericity gave a value below 0.05, then the chart would be followed and results of other tests would be used.

10.4.1 Task 1 - Speed

The first task involved the user finding a post from Twitter and commenting on it with the phrase 'I love this'. The mean time taken for each of the method was calculated.

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	67.7816	20.07951	10
Medium	58.1609	15.44856	10
Fast	57.7591	15.88954	10
Alternative	98.9595	22.50473	10

Figure 39: Mean of time taken for each method for Task 1.

In the hypothesis, it was theorised as the dwell time decreased the average time to complete task would also go down, but as figure 39 shows the time does not decrease between medium and fast however there were differences in the other values.

Mauchly's Test of Sphericity[a]

Measure: Task1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Epsilon[b] Huynh-Feldt	Epsilon[b] Lower-bound
Speed	.537	4.800	5	.444	.768	1.000	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: Speed

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 40: Results of Mauchly's Test for Task 1 - Speed.

The Mauchly's test (see figure 40) produced a value of 0.444, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Speed	Sphericity Assumed	11318.092	3	3772.697	13.576	.000
	Greenhouse-Geisser	11318.092	2.303	4913.533	13.576	.000
	Huynh-Feldt	11318.092	3.000	3772.697	13.576	.000
	Lower-bound	11318.092	1.000	11318.092	13.576	.005
Error(Speed)	Sphericity Assumed	7503.164	27	277.895		
	Greenhouse-Geisser	7503.164	20.731	361.928		
	Huynh-Feldt	7503.164	27.000	277.895		
	Lower-bound	7503.164	9.000	833.685		

Figure 41: Within Effects for Task 1 - Speed

The sphericity assumed value had a significance value of .000 (see figure 41). This is below 0.05, therefore, the difference between the means is statistically significant, and the null hypothesis H_0A was rejected for the first task of commenting on a post.

The result was as expected, apart from the fast selection not yielding quicker results.

10.4.2 Task1 - Accuracy

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	90.0000	21.08185	10
Medium	70.6190	26.61943	10
Fast	39.2203	36.90370	10
Browser	89.8000	16.42356	10

Figure 42: Mean of accuracy for each method for Task1.

There were little changes in the accuracy of the Slow and Browser values. However, it appears that the accuracy fell as the dwell time to make a selection decreased (see figure 42).

Mauchly's Test of Sphericity[a]

Measure: Task1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Huynh-Feldt	Lower-bound
factor1	.179	13.268	5	.022	.628	.789	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 43: Results of Mauchly's Test for Task 1 - Accuracy.

The Mauchly's test (see figure 43) produced a value of 0.022, meaning the Sphericity assumed value was not be checked for significance. Therefore the value for Greenhouse-Geisser was checked for a value below 0.75. The value that was produced was 0.628, which meant the Greenhouse-Geisser value was checked in the within-subjects effects for significance.

Tests of Within-Subjects Effects

Measure: Task1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	17165.839	3	5721.946	8.698	.000
	Greenhouse-Geisser	17165.839	1.885	9108.295	8.698	.003
	Huynh-Feldt	17165.839	2.368	7250.291	8.698	.001
	Lower-bound	17165.839	1.000	17165.839	8.698	.016
Error(factor1)	Sphericity Assumed	17761.400	27	657.830		
	Greenhouse-Geisser	17761.400	16.962	1047.145		
	Huynh-Feldt	17761.400	21.308	833.537		
	Lower-bound	17761.400	9.000	1973.489		

Figure 44: Within Effects for Task 1 - Accuracy.

The Greenhouse-Geisser value had a significance value of .003, which is below 0.05 (see figure 44). Therefore, the difference between the means is statistically significant, and the null hypothesis H_0B and H_0C were rejected for the first task of commenting on a post.

10.4.3 Task 2 - Speed

The second task involved the user sharing a post from Reddit about Technology to both platforms. The mean time taken for each of the methods was calculated first. The mean time to complete the task was taken in the same way as task 1.

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	50.6429	22.90857	10
Medium	27.2520	16.32511	10
Fast	31.9168	22.93572	10
Alternative	61.2278	17.98486	10

Figure 45: Mean of time taken for each method for Task 2.

The same pattern occurs for speed In Task 2. The time does not decrease between medium and fast however there are differences in the other values (see figure 45).

Mauchly's Test of Sphericity[a]

Measure: Task2

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Epsilon[b] Huynh-Feldt	Epsilon[b] Lower-bound
factor1	.496	5.420	5	.370	.739	.991	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 46: Results of Mauchly's Test for Task 2 - Speed.

The Mauchly's test (see figure 46) produced a value of 0.370, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task2

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	7612.734	3	2537.578	8.195	.000
	Greenhouse-Geisser	7612.734	2.216	3435.357	8.195	.002
	Huynh-Feldt	7612.734	2.972	2561.754	8.195	.001
	Lower-bound	7612.734	1.000	7612.734	8.195	.019
Error(factor1)	Sphericity Assumed	8360.119	27	309.634		
	Greenhouse-Geisser	8360.119	19.944	419.181		
	Huynh-Feldt	8360.119	26.745	312.584		
	Lower-bound	8360.119	9.000	928.902		

Figure 47: Within Effects for Task 2 - Speed.

The sphericity assumed value had a significance value of .000, which is below 0.05 (see figure 47). Therefore, the difference between the means is statistically significant, and the null hypothesis H0A was rejected for the first task of sharing a post.

10.4.4 Task2 - Accuracy

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	88.7500	23.89938	10
Medium	81.4396	32.05178	10
Fast	59.5131	37.77549	10
Browser	89.8000	16.42356	10

Figure 48: Mean of accuracy for each method for Task2.

There were little changes in the accuracy of the Slow, Medium and Browser values. However, it appears that the accuracy fell when the dwell time was changed to fast (see figure 48).

Mauchly's Test of Sphericity[a]

Measure: Task2

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Epsilon[b] Huynh-Feldt	Epsilon[b] Lower-bound
factor1	.597	3.989	5	.554	.785	1.000	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 49: Results of Mauchly's Test for Task 2 - Accuracy.

The Mauchly's test (see figure 49) produced a value of 0.544, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task2

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	5943.275	3	1981.092	2.820	.058
	Greenhouse-Geisser	5943.275	2.355	2523.832	2.820	.075
	Huynh-Feldt	5943.275	3.000	1981.092	2.820	.058
	Lower-bound	5943.275	1.000	5943.275	2.820	.127
Error(factor1)	Sphericity Assumed	18964.843	27	702.402		
	Greenhouse-Geisser	18964.843	21.194	894.832		
	Huynh-Feldt	18964.843	27.000	702.402		
	Lower-bound	18964.843	9.000	2107.205		

Figure 50: Within Effects for Task 2 - Accuracy.

The sphericity assumed value had a significance value of 0.058, which is above 0.05 (see figure 50). Therefore the null hypothesis H_0B and H_0C could not be rejected, meaning that there is no difference between the different methods accuracy for sharing a post.

10.4.5 Task 3 - Speed

The third task involved the user liking a post from Reddit and Twitter in either order. The mean time taken for each of the methods was calculated. The mean time to complete the task was taken in the same way as task 1.

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	16.0327	5.94231	10
Medium	20.2197	13.26643	10
Fast	14.8178	8.80459	10
Alternative	61.3442	21.68482	10

Figure 51: Mean of time taken for each method for Task 3.

The same pattern for the time of the previous two tasks does not appear in Task 3. The time value is similar for SocialEye's different dwell settings, but there was a difference between SocialEyes and the Browser (see figure 51).

Mauchly's Test of Sphericity[a]

Measure: Task3

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Epsilon[b] Huynh-Feldt	Epsilon[b] Lower-bound
factor1	.086	18.986	5	.002	.625	.784	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 52: Results of Mauchly's Test for Task 3 - Speed.

The Mauchly's test produced a value of 0.02, meaning the Sphericity assumed value was not be checked for significance (see figure 52). Therefore the value for Greenhouse-Geisser was checked for a value below 0.75. The value that was produced was 0.625, which meant the Greenhouse-Geisser value was checked in the within-subjects effects for significance.

Tests of Within-Subjects Effects

Measure: Task3

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	14893.108	3	4964.369	37.301	.000
	Greenhouse-Geisser	14893.108	1.876	7940.447	37.301	.000
	Huynh-Feldt	14893.108	2.352	6332.323	37.301	.000
	Lower-bound	14893.108	1.000	14893.108	37.301	.000
Error(factor1)	Sphericity Assumed	3593.426	27	133.090		
	Greenhouse-Geisser	3593.426	16.880	212.876		
	Huynh-Feldt	3593.426	21.167	169.763		
	Lower-bound	3593.426	9.000	399.270		

Figure 53: Within Effects for Task 3 - Speed.

The Greenhouse-Geisser value had a significance value of 0.000, which was below 0.05 (see figure 53). Therefore, the difference between the means is statistically significant, and the null hypothesis H_0A was rejected for the third task of liking a post.

10.4.6 Task3 - Accuracy

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	76.9769	30.58191	10
Medium	54.2940	35.39200	10
Fast	37.5651	31.55312	10
Browser	88.6000	15.83386	10

Figure 54: Mean of accuracy for each method for Task3.

There were little changes in the accuracy of the Slow and Browser values. However, it appears that the accuracy fell as the dwell time to make a selection decreased (see figure 54).

Mauchly's Test of Sphericity[a]

Measure: Task3

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Epsilon[b] Huynh-Feldt	Epsilon[b] Lower-bound
factor1	.433	6.469	5	.266	.650	.828	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 55: Results of Mauchly's Test for Task 3 - Accuracy

The Mauchly's test (see figure 55) produced a value of 0.266, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task3

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	15660.561	3	5220.187	6.902	.001
	Greenhouse-Geisser	15660.561	1.951	8025.148	6.902	.006
	Huynh-Feldt	15660.561	2.485	6302.513	6.902	.003
	Lower-bound	15660.561	1.000	15660.561	6.902	.027
Error(factor1)	Sphericity Assumed	20421.708	27	756.360		
	Greenhouse-Geisser	20421.708	17.563	1162.774		
	Huynh-Feldt	20421.708	22.363	913.179		
	Lower-bound	20421.708	9.000	2269.079		

Figure 56: Within Effects for Task 3 - Accuracy.

The sphericity assumed value had a significance value of .001, which is below 0.05 (see figure 56). Therefore, the difference between the means is statistically significant, and the null hypothesis H_0B and H_0C were rejected for the third task of liking a post.

10.4.7 Task 4 - Speed

Finally, task 4 involved the user finding a food related post, stating the food and ending the program/liking a post, both of which indicated when the run of the study ended.

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	57.8987	35.26830	10
Medium	32.9069	10.08710	10
Fast	36.1721	15.19259	10
Alternative	82.3848	39.82912	10

Figure 57: Mean of time taken for each method for Task 4.

The same pattern occurs for speed In Task 2. The time does not decrease between medium and fast however there are differences in the other values (see figure 57).

Mauchly's Test of Sphericity[a]

Measure: Task4

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Huynh-Feldt	Lower-bound
factor1	.120	16.394	5	.006	.484	.553	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 58: Results of Mauchly's Test for Task 4 - Speed.

The Mauchly's test (see figure 58) produced a value of 0.006, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task4

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	15726.366	3	5242.122	6.277	.002
	Greenhouse-Geisser	15726.366	1.453	10824.734	6.277	.018
	Huynh-Feldt	15726.366	1.660	9473.824	6.277	.014
	Lower-bound	15726.366	1.000	15726.366	6.277	.034
Error(factor1)	Sphericity Assumed	22550.189	27	835.192		
	Greenhouse-Geisser	22550.189	13.075	1724.632		
	Huynh-Feldt	22550.189	14.940	1509.401		
	Lower-bound	22550.189	9.000	2505.577		

Figure 59: Within Effects for Task 4 - Speed.

The sphericity assumed value had a significance value of .002, which is below 0.05 (see figure 59). Therefore, the difference between the means is statistically significant, and the null hypothesis H_0A was rejected for the fourth task of navigating the central feed.

10.4.8 Task4 - Accuracy

Descriptive Statistics

	Mean	Std. Deviation	N
Slow	68.3732	40.93049	10
Medium	68.7429	29.86628	10
Fast	45.9674	39.45534	10
Browser	63.5000	39.44405	10

Figure 60: Mean of accuracy for each method for Task4.

There were little changes in the accuracy of the Slow, Medium and Browser values. However, it appears that the accuracy fell when the dwell time was changed to fast (see figure 60).

Mauchly's Test of Sphericity[a]

Measure: Task4

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Huynh-Feldt	Lower-bound
factor1	.592	4.044	5	.546	.757	1.000	.333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: factor1

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figure 61: Results of Mauchly's Test for Task 4 - Accuracy.

The Mauchly's test (see figure 61) produced a value of 0.546, meaning the Sphericity assumed value was checked for significance.

Tests of Within-Subjects Effects

Measure: Task4

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	3448.762	3	1149.587	.817	.496
	Greenhouse-Geisser	3448.762	2.271	1518.786	.817	.470
	Huynh-Feldt	3448.762	3.000	1149.587	.817	.496
	Lower-bound	3448.762	1.000	3448.762	.817	.390
Error(factor1)	Sphericity Assumed	37987.329	27	1406.938		
	Greenhouse-Geisser	37987.329	20.437	1858.787		
	Huynh-Feldt	37987.329	27.000	1406.938		
	Lower-bound	37987.329	9.000	4220.814		

Figure 62: Within Effects for Task 4 - Accuracy.

The sphericity assumed value had a significance value of 0.496, which is above 0.05 (see figure 62). Therefore the null hypothesis H_0B and H_0C could not be rejected, meaning that there was no difference between the different methods accuracy for navigating the central feed.

10.5 Questionnaire Feedback

Q1) How easy to learn was the user interface for the SocialEyes application?

Q2) How easy to learn was the user interface for the currently used application?

Q3a) How easy did you find 'liking' a post using the SocialEyes application?

Q3b) How easy did you find 'liking' a post using the currently used application?

Q4a) How easy did you find 'commenting' on a post using the SocialEyes application?

Q4b) How easy did you find 'commenting' on a post using the currently used application?

Q5a) How easy did you find 'sharing' a post using the SocialEyes application?

Q5b) How easy did you find 'sharing' a post using the currently used application?

The analysis performed previously was parametric. However, this cannot run over ranked or ordinal data such as the Likert scale in the questionnaire, due to the scale not being linear in its values. Therefore a nonparametric test was performed instead to check for significant results in the questionnaire comparing the two systems. SPSS was used to run a Mann Whitney test over the data.

10.5.1 How easy was the user interface to learn?

Mann-Whitney Test

Ranks

	Group	N	Mean Rank	Sum of Ranks
Score	SocialEyes	10	11.90	119.00
	Web Based	10	9.10	91.00
	Total	20		

Test Statistics[a]

	Score
Mann-Whitney U	36.000
Wilcoxon W	91.000
Z	-1.105
Asymp. Sig. (2-tailed)	.269
Exact Sig. [2*(1-tailed Sig.)]	.315[b]

a. Grouping Variable: Group

b. Not corrected for ties.

Figure 63: Mann-Whitney score for the usability of systems.

In the Mann-Whitney test, the mean rank is analysed rather than the mean score, as rank data cannot have algorithms such as mean applied to them. The mean rank for SocialEyes was higher than the web-based system (see figure 63). However, the significance score is .269, which is not below 0.05 meaning we cannot state that there was a significant difference in learning the user interface. Both interfaces scored well enough to show that they are easy to learn, but

SocialEyes was not more straightforward to learn compared to the web-based alternative.

10.5.2 How easy was it to like a post?

Mann-Whitney Test

Ranks

	Group	N	Mean Rank	Sum of Ranks
Score	SocialEyes	10	14.40	144.00
	Web based	10	6.60	66.00
	Total	20		

Test Statistics[a]

	Score
Mann-Whitney U	11.000
Wilcoxon W	66.000
Z	-2.987
Asymp. Sig. (2-tailed)	.003
Exact Sig. [2*(1-tailed Sig.)]	.002[b]

a. Grouping Variable: Group

b. Not corrected for ties.

Figure 64: Mann-Whitney score for liking a post using the systems.

For the Mann-Whitney test (see figure 64) for liking a post using the system, there is a significant result, and there was a difference in the two systems due to the significance score of 0.03 being less than 0.05. When looking at the mean rank SocialEyes score was much higher than the web-based version, showing that SocialEyes was easier to like posts with.

10.5.3 How easy was it to comment on a post?

Mann-Whitney Test

Ranks

	Group	N	Mean Rank	Sum of Ranks
Score	SocialEyes	10	13.35	133.50
	Web based	10	7.65	76.50
	Total	20		

Test Statistics[a]

	Score
Mann-Whitney U	21.500
Wilcoxon W	76.500
Z	-2.207
Asymp. Sig. (2-tailed)	.027
Exact Sig. [2*(1-tailed Sig.)]	.029[b]

a. Grouping Variable: Group

b. Not corrected for ties.

Figure 65: Mann-Whitney score for commenting on a post using the systems.

The same Mann-Whitney test (see figure 65) ran for commenting on a post, with there also being a significant difference in the mean rank score of the two systems. A significance score or 0.027 shows that there was a significant difference in the two systems. SocialEyes was easier to use when commenting on posts.

10.5.4 How easy was it to share a post?

Mann-Whitney Test

Ranks

	Group	N	Mean Rank	Sum of Ranks
Score	SocialEyes	10	14.35	143.50
	Web based	10	6.65	66.50
	Total	20		

Test Statistics[a]

	Score
Mann-Whitney U	11.500
Wilcoxon W	66.500
Z	-2.977
Asymp. Sig. (2-tailed)	.003
Exact Sig. [2*(1-tailed Sig.)]	.002[b]

a. Grouping Variable: Group

b. Not corrected for ties.

Figure 66: Mann-Whitney score for sharing a post using the systems.

Finally, the Mann-Whitney test (see figure 66) ran for sharing a post. Following on from the previous tests there was a significance score of 0.03, showing there was a significant difference between the two systems. SocialEyes mean rank was higher, therefore showing that it was easier to share posts with than the web-based alternative.

Overall the Likert scale questions showed that SocialEyes was easier to use for all of the common features found in social media applications compared to the current web-based alternative.

10.5.5 Written Feedback

Each participant also answered two questions about how SocialEyes could be improved and any other general comments about SocialEyes.

There was a suggestion to make the scrolling of the page more fluid, similar to how a web page works. Another suggestion was to improve the overall aesthetics of the system while also making the scrolling buttons and icons slightly larger. The main improvement that was said by a third of users was to have a neutral space where no dwell actions would occur, so when making a selection

they were not instantly dwelling on an option.

There were other comments about the application, 20 percent of users found the fast speed was too quick, causing them to make more mistakes.

However, there were lots of positive comments as well for the application, with 20 percent of users stating that the colour of the tags made it much easier to find posts of a particular type. Multiple users also stated that the larger buttons made selections much more accessible, also saying that it was much better than the web application. Users also stated that it was easy to learn, the selection was easy, and that medium was the best option to use on balance. Most importantly multiple users stated they believed this would significantly improve the lives of disabled users by allowing them to have an online experience.

10.6 Final Experimental Report

The original hypothesis for the experiment was to show that the time taken would decrease as the dwell time did, with all the dwell times being quicker than the web alternative. For all four tasks that were performed during the user study H_0A was rejected, showing significance for the tasks when analysing speed. There was minimal difference between the medium and fast dwell times, but time taken does decrease between browser to slow and then slow to medium/fast.

Another original hypothesis was accuracy decreased as the dwell time decreases. The user study produced mixed results, with Task 1 and Task 3 rejecting H_0B, while Task 2 and Task 4 were unable to reject H_0B. For the Tasks that rejected H_0B, the accuracy consistently fell as the dwell time decreased. However, in Task 2 and Task 4 the accuracy for the fast dwell time was considerably lower, which correlates with the users' written feedback since the users in the written feedback stated the they had issues using the fast dwell time.

The Mann-Whitney tests showed that SocialEyes was easier to use for the common tasks assessed during the study, but it was not significantly easier to learn. These tests backup the ANOVA's results where it showed that SocialEyes was significantly quicker to use for all the assessed tasks.

10.6.1 Reflection on Results

After looking back on the results of the experiments and the feedback on the questionnaires, I believe some of the hypothesis results did not meet what was expected. Initially, I thought that as the speed increased the user would make quicker selections, but make more frequent mistakes due to not being as familiar with the system. However, as shown during the redesign of the experiment users picked up the slow and medium selection methods well, with the accuracy only

slightly decreasing. The score for how quickly a user learned to use SocialEyes reflects this. However, the fastest speed was not another equal jump, which was due to it being so fast it became hard to use and error-prone, which is demonstrated in the ANOVA on accuracy and general comments about the system.

The accuracy did not always decrease which is visible in Task 2 and Task 4, where the null hypothesis were not rejected. This task involved many traversals of news feeds, and results show that users found the tagging of items useful for finding and identifying tasks, which backed up the accuracy results for Task 4.

The accuracy of the system could potentially increase by introducing rest areas or neutral spaces in the menu selection areas, which was where most users felt like they made their mistakes. A further study could be conducted to test this theory once implemented with the system.

11 Evaluation

This section considers the success of the project in different contexts, First, the system will be evaluated against the primary and secondary project objectives. Then the success of the project as a whole will be looked at along with the experimental results comparing it to the real world alternative to the project.

11.1 Progress of Objectives

11.1.1 Primary Objectives

P1 Conduct research identifying a feature set common amongst social media platforms.

P2 Design and Prototype a system for managing multiple social media platforms using eye tracking as the input.

P3 Create a final application using identified feature set allowing users to interact with items/posts.

P4 Conduct a study comparing the created system against other similar systems, analysing the quantitative and qualitative results.

For objective 1, in-depth research surrounding the history of social media sites and the common features present in them was carried out. A broad sweep of relevant papers and articles online detailing the development of social media were collected. Any common themes were collated, with many features that were specific to a platform discounted. If a common feature was feasible for the system, it was selected and researched further to see how it could affect the system.

Objective 2 was completed after objective 1, with a full HCI design process followed. Low-fidelity prototyping was performed via paper prototypes. Feedback was gathered on and designs were revised too. A storyboard following the previous ideas then led the creation of a mock application. The feedback and changes gained through the process allowed a system which handled multiple social media platforms to be prototyped.

Objective 3 followed the completion of objective 2. A full application was created that allowed users to complete the full set of common features of; like, comment, share and posting. Twitter and Reddit APIs handled the features, with them called through the use of the Tobii EyeX tracker.

Objective 4 was the last objective to be completed, with the study relying on the system created. A within-subject study was conducted comparing SocialEyes with the current web-based alternative. Quantitative comparison was done when time and accuracy were recorded and then compared with an ANOVA where the significant results were reported. Qualitative results were

achieved through the questionnaire, with all significant results and other feedback reported.

11.1.2 Secondary Objectives

S1 Allow users to filter items in the central feed.

S2 Allow users to customise which features from the feature set they want to use.

S3 Have the UI adapt to filtered/customisable changes the user chooses.

Objective 1 was completed. In the UI a menu was created which allowed users to select which categories of posts and which social media platforms they wish to see inside the central feed. The filtering of the posts worked, with only posts with tags that the user wanted in the feed being in the feed. The objective was completed by the program assigning tags to posts. A basic logic operation was applied to posts, which offered accurate but not perfect results. Machine learning was applied to the posts using NLP, but it was not fully integrated with the system.

Objective 2 was completed with the use of the action filter menu. With the concept of radial menus meaning that the fewer options in the menu the more substantial the selection area per option, which meant that when the user customised to only have the features they wanted they could select those features more quickly.

Objective 3 was partially completed. The UI would update the central feed to any filtering that was applied changing which posts and features were on offer to the user, giving informative feedback to the user. More aspects of the UI could have been customisable but were not, such as being able to choose the tags available, change the filtering menu to only filter what users wanted to filter as well as the size of the posts in the news feed.

11.2 Reflection on Objectives

As the project continued the set of objectives initially decided on evolved and were broken down. The third primary objective could have been broken down into two tasks of creating an interface and handling social media data. The third secondary objective could have been explained more and split to more specific changes to particular sections of the UI.

11.3 Evaluation against Current Systems

Considering how the project compares to current research and systems is essential in completing an evaluation of the project. The main comparison with

current systems was the entire concept of running a user study. The results at the end of the study empirically showed how SocialEyes compared with the currently used application, with all significant results reported. It highlighted how SocialEyes addressed the accuracy and Midas touch problems associated with eye tracking. SocialEyes had settings which made users quicker and more accurate, while the users gave meaningful feedback through the questionnaire showing that it was better to use. SocialEyes was not significantly easier to learn than the other application, but due to it being an unusual system input and still scoring high in the scores for easiness to learn it is not a significant issue.

The pie menu proposed by Majaranta et al.[14] was one of the examples analysed in the context survey. The speed scores achieved for the port-hole pie menu were lower than expected, but SocialEyes was able to reach a speed where users felt it could not go faster. In this project, the accuracy was also assessed, which Majaranta et al. failed to do. They did allow users to see what they were working on, which helped produce the ability to see posts in full and position the text boxes in locations where the keyboard would not obscure them.

Another of the examples by Zhang, Yao and Cai[42] was the nine directional text input. With the concept based on the classic T9 input, which allowed users to have large areas to select. However, this is restrictive to what appears in a menu system, but in SocialEyes the menus can be changed to allow the user to decide how the menu is set up.

11.4 Limitations and Improvements

There are few limitations with the system, for example, the set social media applications that are attached to the system. With only Twitter and Reddit currently available on the system. In future, it would be ideal for the user to select from a list of social media platforms that support the common feature set, which would add another level of customisability to the system. With the limited time scale of the project, it was not possible to complete all procedures of obtaining social media permissions, but again this could be achieved in later iterations of the project.

When analysing the feedback sections of the questionnaire, comments were describing the aesthetics of the system. The layout and event handlers for the project were designed from scratch. Therefore there were limitations to how advance the graphics would become. The overall aesthetics of the system could improve in future, which would again enhance the system for the user.

Currently in order to obtain data from the Reddit API an OAuth generator based on a web page is used. When supplied with the application id and secret key, the authorisation codes are produced, but they only last for forty-five minutes and need to be regenerated again after this period. This process should be automated within the system, but this was not possible in the time scale of the project. The login section would be a priority in future releases of the system. The automation of the generation could pair with a login area for the system where the user would attach their accounts. Currently, users accounts are hardcoded into the system.

Another piece of general feedback in the questionnaires was the placement of the submit comment button, which would be dwelled on immediately after the selection of the comment command from the react menu. A redesign would need to take place, in order to make sure users landed on a no dwell zone after selecting a menu item, which would appear in a future iteration of the system.

When using the different dwell times in SocialEyes during the user study, they were adjusted to the different speeds much quicker than expected. Three different dwell times were provided due to the idea that it would take a long time for users to adjust to the speeds. A future improvement would be a draggable menu where the exact dwell time could be set in the system, allowing the user full customisability over the dwelling problem of the system.

More refactoring could occur on the system. Since one of the main goals of the system was to conduct a user study, the other functionality of the system had to be completed beforehand, and this was easiest with code centralised in one file. With the small time frame of the project, there was limited time to complete full refactoring of the system, which again would be completed if more time was available on the project.

As mentioned previously in the report, the filtering of the system, specifically the machine learning being included with the main functionality of the project. The machine learning currently works separately. The logical tagging implemented instead was functional and served the system well for proving the system, but the machine learning paired with NLP could in future offer very accurate real-time tagging with various new categories.

Limited testing was completed during the project, with some minor issues detected. When the system was loading the news feed on launch or during the feed being filtered/refreshed, some menus can have accidental dwell on them. The dwell time starts to count even though the menu is not visible on the screen.

There were issues around the API calls and how often it can occur. There is a limited number of items that can be retrieved by social media platforms in a certain period. Therefore if the system is refreshed and filtered repeated the system will timeout and the user will have to wait until the API allows calls again, which is usually approximately fifteen minutes, which could be rectified with paid subscriptions, but this was not possible during the project.

12 Conclusion

12.1 Overview of Objectives

The main objective of the project was to create a social media collaboration tool designed for disabled users, with eye tracking as the only inputs, which would improve the overall experience for a disabled user, aiding them in their physical and mental health. This is represented by the primary objectives of the project, where social media platforms were researched, the system was designed thoroughly, with it then fully implemented and a user study performed showing where SocialEyes improved against the current alternative. All of these objectives were successfully met. There were also custom menus and features making the application more natural to use.

12.2 Future Works

The system can be expanded to add the additional features discussed in the previous sections. More social media platforms are an option, which could then aid in the expansion of adding more categories to the system.

The main piece of work which could be carried out in future would be a more in-depth study investigating how it benefits disabled users. A revised system with the previous improvements implemented would be tested with the target group of users. People that currently use social media with eye tracking could be assessed about how it makes them feel while using, with them then using SocialEyes for an extended period of time, measuring how the system makes them feel to see if the system could help users to deal with the physical and mental effects of their disability. Extensive ethical consideration would need to be carried out due to the unique nature of the participants required for the study.

12.3 Personal Reflection

I feel this project has been a success. I have produced a social media collaboration tool with eye tracking as the only input, with a full user study completed comparing it to a current alternative, which showed SocialEyes results were significant. The system created and the results gained during the study show that the application had the possibility to improve disabled users interactions with social media. I feel the system has a lot of potential and with the improvements mentioned could help many users.

On a personal level I have learned throughout. This was the first time I was required to complete this scale of project in the period of time completed. This allowed me to learn how to prioritise and make sure an implementation was completed on time, even with the deadline being changed during the project.

This was important as it allowed me to learn how to design and run a user study. I also discovered the importance of a pilot study and how it could lead to a more detailed user study being implemented. Finally, I learned how to execute a full HCI design process, seeing how the project can benefit from it. The whole design process is one I will use in future when creating an application.

I decided to propose a project that involved using a piece of hardware I had no experience with, which in this case was an eye tracker. As the project reached the end, I feel I am fully aware and able to work with the issues that arise when working with an eye tracker and would be able to use an eye tracker in future without worries.

Finally, this was the first large project with a study that I have completed, and the experience I have gained from it will be valuable when developing future research projects.

12.4 Acknowledgements

I want to thank my supervisor Ruth Letham for another year of help and support while completing my final university project. I would also like to thank my high school teacher Mr Grewar who served as the inspiration for the project and helped me reach where I have in my life.

13 Appendix

13.1 A - Ethics CS12476

UNIVERSITY OF ST ANDREWS
TEACHING AND RESEARCH ETHICS COMMITTEE (UTREC)
SCHOOL OF COMPUTER SCIENCE
ARTIFACT EVALUATION FORM

Title of project

SocialEyes: Enhancing SocialMedia Collaboration for Disabled Users, with Eye Tracking

Name of researcher(s)

Christopher Fleming

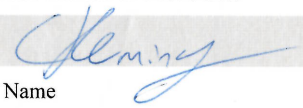
Name of supervisor

Dr Ruth Letham

Self audit has been conducted **YES** ☒ **NO** ☐

This project is covered by the ethical application CS12476

Signature Student or Researcher



Print Name

CHRISTOPHER FLEMING

Date

12/02/2019

Signature Lead Researcher or Supervisor

Print Name

RUTH LETHAM

Date

12/02/2019

13.2 B - Ethics CS14257

University of St Andrews | FOUNDED 1413

University Teaching and Research Ethics Committee

13 May 2019

Dear Christopher,

Thank you for submitting your ethical application, which was considered by the School of Computer Science Ethics Committee on Wednesday 16th January, where the following documents were reviewed:

1. Ethical Application Form
2. Participant Information Sheet
3. Consent Form
4. Debriefing Form
5. Advertisement Form

The School of Computer Science Ethics Committee has been delegated to act on behalf of the University Teaching and Research Ethics Committee (UTREC) and has granted this application ethical approval. The particulars relating to the approved project are as follows -

Approval Code:	CS14257	**Approved on:**	26.04.19	**Approval Expiry:**	26.04.2024
Project Title:	SocialEyes: Enhancing Social Media for disabled users				
Researcher(s):	Christopher Fleming				
Supervisor(s):	Ruth Letham				

Approval is awarded for five years. Projects which have not commenced within two years of approval must be re-submitted for review by your School Ethics Committee. If you are unable to complete your research within the five year approval period, you are required to write to your School Ethics Committee Convener to request a discretionary extension of no greater than 6 months or to re-apply if directed to do so, and you should inform your School Ethics Committee when your project reaches completion.

If you make any changes to the project outlined in your approved ethical application form, you should inform your supervisor and seek advice on the ethical implications of those changes from the School Ethics Convener who may advise you to complete and submit an ethical amendment form for review.

Any adverse incident which occurs during the course of conducting your research must be reported immediately to the School Ethics Committee who will advise you on the appropriate action to be taken.

Approval is given on the understanding that you conduct your research as outlined in your application and in compliance with UTREC Guidelines and Policies (http://www.st-andrews.ac.uk/utrec/guidelinespolicies/). You are also advised to ensure that you procure and handle your research data within the provisions of the Data Provision Act 1998 and in accordance with any conditions of funding incumbent upon you.

Yours sincerely

Wendy Boyter

School Ethics Committee Administrator

ethics-cs@st-andrews.ac.uk

13.3 C - Questionnaire

Sample Questionnaire for Study

1. How easy to learn was the user interface for the SocialEyes application?

   1 2 3 4 5 6 7 8 9 10

2. How easy to learn was the user interface for the currently used application?

   1 2 3 4 5 6 7 8 9 10

3.a) How easy did you find ‘liking’ a post using the SocialEyes application?

   1 2 3 4 5 6 7 8 9 10

b) How easy did you find ‘liking a post using the currently used application?

   1 2 3 4 5 6 7 8 9 10

4.a) How easy did you find ‘commenting’ on a post using the SocialEyes application?

   1 2 3 4 5 6 7 8 9 10

b)How easy did you find ‘commenting’ on a post using the currently used application?

1 2 3 4 5 6 7 8 9 10

5.a) How easy did you find ‘sharing’ a post using the SocialEyes application?

1 2 3 4 5 6 7 8 9 10

b)How easy did you find ‘sharing’ a post using the currently used application?

1 2 3 4 5 6 7 8 9 10

6.a) How easy did you find ‘posting’ an item using the SocialEyes application?

1 2 3 4 5 6 7 8 9 10

b) How easy did you find ‘posting’ an item using the currently used application?

1 2 3 4 5 6 7 8 9 10

7. What aspects of the SocialEyes application could be improved?

8. Any general comments surrounding the SocialEyes application?

References

[1] *CS3106 - Human Computer Interaction, Lecture 4, Quigley,A, 2016.* Accessed on 3/5/19.

[2] *Ellis, K., Goggin, G., Huntsinger, J., Senft, T. (2014). Disability and social media. The social media handbook, 126-143.*

[3] *Facebook Cambridge Analytica Scandal.* Accessed on 21/2/19.

[4] *Gamberini, L., Barresi, G., Maier, A., Scarpetta, F. (2008). A game a day keeps the doctor away: A short review of computer games in mental healthcare. Journal of CyberTherapy and Rehabilitation, 1(2), 127-145.*

[5] *Gibaldi, A., Vanegas, M., Bex, P. J., Maiello, G. (2017). Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research. Behavior research methods, 49(3), 923-946.*

[6] *HAR Analyzer.* Accessed on 1/5/19.

[7] *IBM SPSS.* Accessed on 9/5/19.

[8] *Icons sources.* Accessed on 21/2/19.

[9] *Istance, H., Bates, R., Hyrskykari, A., Vickers, S. (2008, March). Snap clutch, a moded approach to solving the Midas touch problem. In Proceedings of the 2008 symposium on Eye tracking research applications (pp. 221-228). ACM.*

[10] *Kaplan, A. M., Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. Business horizons, 53(1), 59-68.*

[11] *Key Generator.* Accessed on 10/4/19.

[12] *Kumar, M., Paepcke, A., Winograd, T., Winograd, T. (2007, April). EyePoint: practical pointing and selection using gaze and keyboard. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 421-430). ACM.*

[13] *Lemahieu, W., Wyns, B. (2011). Low cost eye tracking for human-machine interfacing.*

[14] *Majaranta, P., Majaranta, N., Daunys, G., Špakov, O. (2009, May). Text editing by gaze: Static vs. dynamic menus. In In Proceedings of the 5th Conference on Communication by Gaze Interaction; COGAIN (pp. 19-23).*

[15] *Microsoft assistance bar.* Accessed on 9/5/19.

[16] *MineCraft Community.* `https://feedback.minecraft.net/hc/en-us/articles/360004822172-Official-Minecraft-Community-Realm`. Accessed on 18/3/19.

[17] *Mockplus.* Accessed on 1/5/19.

[18] *Munzner, T. (2014). Visualization analysis and design. AK Peters/CRC Press.*

[19] *Phaser for Games.* Accessed on 9/5/19.

[20] *Piao, G., Jin, Q., Zhou, X., Nishimura, S., Wattanachote, K., Shih, T. K., Yen, N. Y. (2013, December). Eye-tracking experiment design for extraction of viewing patterns in social media. In 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing (pp. 308-313). IEEE.*

[21] *Piazolo, F., Geist, V., Brehm, L., Schmidt, R. (Eds.). (2017). Innovations in Enterprise Information Systems Management and Engineering: 5th International Conference, ERP Future 2016-Research, Hagenberg, Austria, November 14, 2016, Revised Papers (Vol. 285). Springer.*

[22] *Processing for Games.* Accessed on 9/5/19.

[23] *Prototyping for Design and Evaluation. Greenberg, S.* `http://grouplab.cpsc.ucalgary.ca/saul/681/1998/prototyping/survey.html#storyboard`. Accessed on 29/4/19.

[24] *Reddit.* `https://www.reddit.com/wiki/reddiquette/`. Accessed on 29/4/19.

[25] *Reddit C Sharp Library.* Accessed on 2/5/19.

[26] *Repeated Measures ANOVA - Report Which Results?* Accessed on 3/5/19.

[27] *Restricted Facebook API.* Accessed on 3/5/19.

[28] *Rudd, J., Stern, K., Isensee, S. (1996). Low vs. high-fidelity prototyping debate. interactions, 3(1), 76-85.*

[29] *Schultz, S. (2013). Social Media as a Communication Tool for Individuals with Disabilities Living in Rural Settings.*

[30] *Second Life.* `https://secondlife.com/`. Accessed on 18/3/19.

[31] *Shneiderman, B., Plaisant, C. (2010). Designing the user interface: strategies for effective human-computer interaction. Pearson Education India.*

[32] *Shneiderman's Eight Golden Rules of Interface Design.* Accessed on 3/4/19.

[33] *Tobii EyeX Specifications.* Accessed on 9/5/19.

[34] *Tweet Data.* Accessed on 19/4/19.

[35] *Twitter C Sharp Library.* Accessed on 2/5/19.

[36] *Twitter Developer Platform.* Accessed on 22/4/19.

[37] *Urbina, M. H., Huckauf, A. (2010, March). Alternatives to single character entry and dwell time selection on eye typing. In Proceedings of the 2010 symposium on eye-tracking research applications (pp. 315-322). ACM.*

[38] *Ward, D. J., Blackwell, A. F., MacKay, D. J. (2000, November). Dasher-a data entry interface using continuous gestures and language models. In UIST (pp. 129-137).*

[39] *Ware, C., Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input2. Acm sigchi bulletin, 18(4), 183-188.*

[40] *Why Icons are Important*. Accessed on 27/4/19.

[41] *Youtube*. `https://youtube.googleblog.com/2009/09/five-stars-dominate-ratings.html`. Accessed on 29/4/19.

[42] *Zhang, C., Yao, R., Cai, J. (2018). Efficient eye typing with 9-direction gaze estimation. Multimedia Tools and Applications, 77(15), 19679-19696.*