Project Title
Interim Report

DT228
BSc in Computer Science

Damian Wojtowicz

C17413722

Andrea Curley

School of Computer Science

Technological University, Dublin

Date

Abstract

The goal of this project is to investigate the use facial expression recognition technology to improve software usability and user experience testing. The application will be of use to software developers who will have users complete test cases and the application will record the user's facial expressions to better detect the problem areas in the software.

This project is going to use machine learning to automatically detect the user's emotions and display the data to the developers for further analysis. The advantage of automatically detecting emotions is that video footage does not have to be watched by anyone and data is automatically collected and stored.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

_____

Student Name

Date

Acknowledgements

Body text

Table of Contents

1. Introduction

1.1.Overview

The purpose of this project is to explore facial expression recognition technology as a way to improve usability testing for software. This project will focus mainly on usability testing on websites. Usability testing is a very important aspect of software development and especially crucial when the application has a large amounts of users with varying degrees of ability. Usability testing involves testing the functionality of an application, a website or digital product by observing how real people with no prior knowledge of the website attempt to navigate through it and attempt to complete tasks [1].

Usability testing reveals issues that are not obvious to developers, designers and people who have in-depth knowledge and understanding of the product. By observing the user, the designer can not only identify problems but also uncover opportunities to improve and learn about the target user's preferences. For a website to be successful it must allow its users to efficiently and effectively get to where they want to be and complete the tasks they want to complete without confusion and frustration.

Research has been conducted on the number of participants needed to reveal the most problems with the best returns. In a study conducted by Virzi (1990, 1992) random samples of different sizes of participants were created, it was found that on average, four or five participants would reveal about 80% of the usability problems uncovered by all the participants, with diminishing returns with additional participants [2]. Faulkner (2003) also concluded that testers may want to include more than five participants. With 100 random samples of five participants, she found that on average, the five-person samples uncovered 85% of the problems [2]. However, the results by in research paper by Bastien *et. al* (2003) [3] indicated that the first 5 participants only uncovered 35% of the usability problems.

The purpose of this project is to implement a website usability testing tool and investigate if usability testing can be improved upon with the use of machine learning and more specifically facial expression recognition.

1.2.Project Description

UsabCheck is usability testing application and consists of two applications. The first application is a web application that will be used by the researcher, they could be the developer or designer of their website. The researcher will access the *UsabCheck* web application to create and configure their usability study. The researcher will provide the link to the website they want to conduct the usability test on and provide instructions for the tasks they want the participant to complete and questionnaires and questions they want the user to fill out at any point within the study.

The *UsabCheck* web application will provide statistics on how participants have performed in the tasks and their responses to questions and other forms. The researcher will have the option to view the screen recording and a camera recording of their face while they are completing the tasks. This includes the audio of the participant speaking. The researcher will also be provided with a journey map of the user completing the task and their facial expressions to help better identify the problem areas.

The second UsabCheck application will run locally on the computer the participant is using and will be recording their screen and monitoring their facial expressions. After the usability test is conducted the data is uploaded to the cloud. This data includes video's and screen captures.

1.3.Project Aims and Objectives

The aims and objectives that should be achieved by the end of the project include the following.

1. Provide a usability testing website that allows the researchers to create usability study tests.
2. Provide the researchers to view the results of the tests on the website. This includes viewing of the footage, answers to questions and questionnaires, viewing a journey map of facial expressions.
3. Provide a local usability testing application that will record the screen, record the test participants face from the camera, show the participant the test instructions, and upload the data to the cloud.
4. For the above aims to be achieved, smaller goals and objectives have to be met. This includes a machine learning algorithm that will classify facial expressions.
5. The machine learning algorithms and dataset have to be researched and implemented.
6. Cloud storage has to be obtained in some way, whether it be creating my own or paying for a service to store the videos for streaming on the website.

1.4.Challenges

1. The dataset for the face expression recognition will be a very important factor in the success of the classification.
2. The various algorithms for face classification need to be researched and considered as this will have an impact on the accuracy of face expression classification.
3. The usability testing aspect has to be researched well for the application to be useful. The scope of this project is big in that regard.
4. There are various ways of going about cloud storage, the pros and cons of each will have to be considered.
5. Time management is a challenge that is very important to the success of the project.

1.5.Project Scope

The scope of this project involves designing and implementing a usability application for desktop applications. More specifically the focus will be on website usability testing. The application will not be intended for usability testing on mobile applications. The application will involve the use of machine learning to detect facial expressions of the participant who is completing the tests created by the researchers or developers. The project will involve two applications; a web application and a local application.

1.6.Thesis Roadmap

Will be added at the very end

2. Research

2.1. Background Research

2.1.1 Emotions in Usability Testing

Landowska (2015) [4] investigated the use of emotion analysis in usability testing. They have evaluated the methods of obtaining the emotion data and the models for representing that data. The means of obtaining the data which include questionnaires, facial expression analysis and sentiment analysis were evaluated by the accuracy, robustness to disturbances, independence on human will, and interference with usability testing procedures. The method of obtaining the emotion recognition data that is relevant to this project is facial expression analysis. The accuracy of this method is medium to high and does not interfere with the usability testing procedure. However, the robustness to disturbances is low, meaning that illumination and occlusion of the face reduces the accuracy of the method.

They have investigated the use of Ekman's six basic emotion model (Figure 1) as a method for representing the emotional state. Examples of emotional states include frustration, boredom, excitement, and empowerment. These emotional states consist of one or more of the more basic emotions such as disgust, anger, surprise, fear, sadness and joy. They have found that with this model certain emotional states are difficult to illustrate with the Ekman's six basic emotion model, for example, boredom can be expressed as subtle sadness which is not the most intuitive representation of the emotional state. Whissel's wheel emotion model (Figure 2) has been investigated. Dimensional models such as Whissel's wheel can be easily interpreted by computer systems, however it is more difficult for humans to understand. Further research on this model can be done to automatically detect more complex emotions in the *UsabCheck* application.

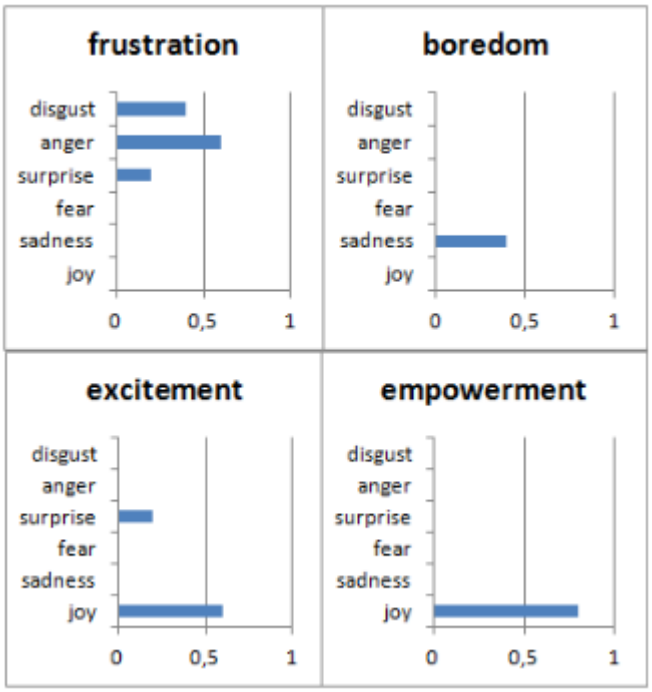


Figure 1

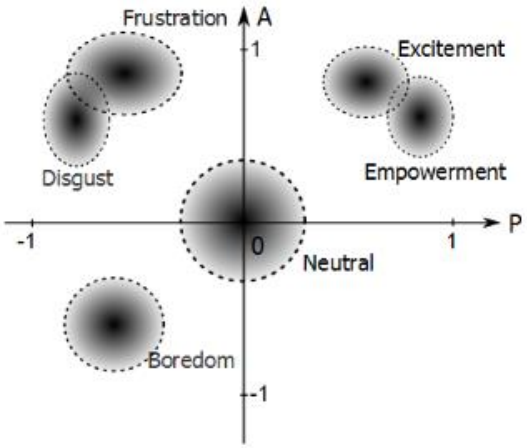


Figure 2

An additional study which involved post hoc analysis of recordings from usability tests was performed. This was to assess the practical applicability of the emotion recognition techniques in a real-world environment. The tests varied in that some participants used a mouse and keyboard and other participants used a touch screen. The instructions were provided on paper and the participants would look away from the camera to look at the instructions. This disturbance meant that the facial expression recognition with the use of the camera was not effective during those times. Some participants covered their face with their hand as much as 33 percent of the time which interfered with the facial expression recognition. The camera was located below the screen.

In conclusion, this research paper provided insight into the various means of detecting emotions and representing the data. The facial expression recognition technique of obtaining data is vulnerable to interference, however, it provides accuracy in detecting the emotion. The 6 basic emotions can be combined in various ways to form more complex emotions such as frustration and boredom. This is certainly useful in providing the researchers who will use the *UsabCheck* application with more intuitive data.

Esterwood (2018) [5] discussed how facial expression recognition technology can be applied to usability testing. They looked at journey maps as a way to display emotion data. Journey maps (Figure 4) track the user as they move through the task and shows the user's emotions along with the task they are trying to complete. In Figure 4, negative emotions of anger and sadness were assigned a -1 value and surprise and happiness were assigned +1 value on the y-axis. The time is plotted on the x-axis and the tasks are separated using a dotted line. The chart plots the emotions in a way that is either positive or negative which is useful. However, it does not display exactly what emotion was experienced which makes this approach debatable. Surprise was marked with a +1 value which is the same as happiness and carries the assumption that the user's surprise is a positive thing. One might argue that for a system to be usable the user should not much experience surprise and user surprise is a result of bad design.

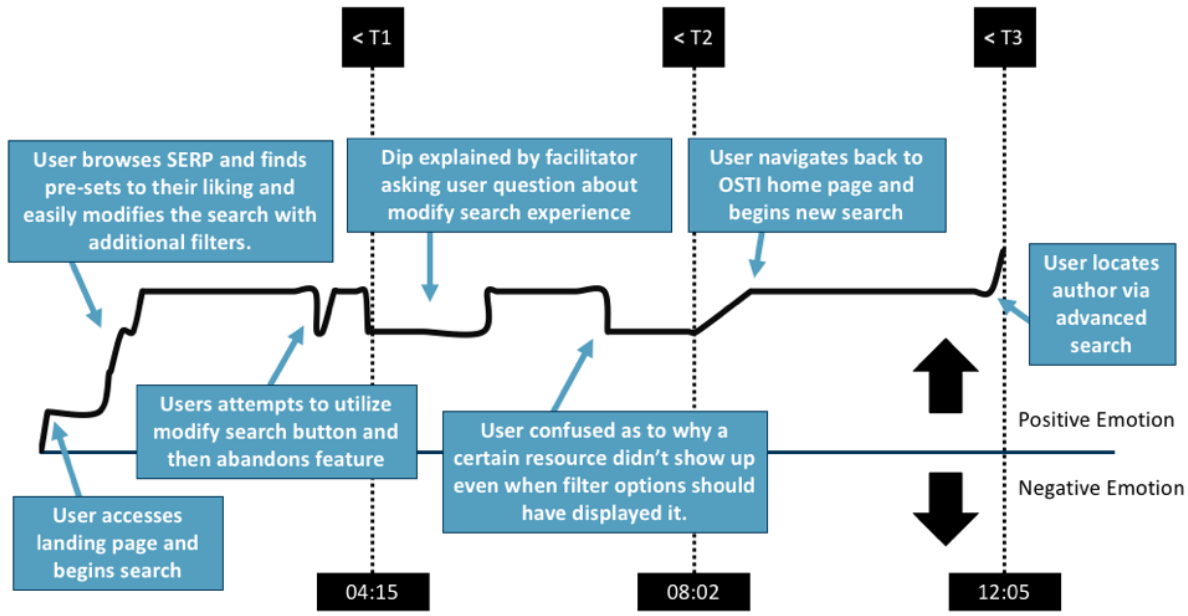


Figure 4

The recommendations by the study for future studies is to ensure that users are limited to a strict time limit as a way to allow for averaging of emotional states and combining them into one journey map. Another recommendation is to have a second camera exclusively for the face as cropping decreases the resolution. In conclusion this study has provided insight into displaying the emotion data in the context of usability testing that is easy to understand. However, the simplicity might be a disadvantage as the chart only works with positive and negative values and the value each emotion should be assigned is up for debate.

The recommendation to limit the amount of time the user is allowed to spend on a task as a way of better averaging the emotional states is very insightful. Users of varying degrees of ability will spend different amount of times on each task and this will certainly be an issue in pinpointing the exact stage the user is during the usability test. Reducing the amount of time, the user can spend on a task can help with that. However, that length of time is subjective to the researcher conducting the usability test.

2.1.2 Facial Expression Recognition Research

Halder et al. (2016) [6] proposed a prototype system which classifies the six basic emotions such as happiness, sadness, anger, fear, surprise, and disgust. They explored a new approach to the emotion classification from facial expression by combing a neural network-based approach with image processing. The system includes face detection and feature extraction for emotion classification. The feature extraction involved locating the eyebrows, eyes and mouth regions. The image processing techniques included the use of edge detection, for example, Sobel edge detection which finds edges by the gradient changes in an image. These features are points and these points are processed to obtain inputs for the neural network. The results of the classification were left out of the research paper since the results were being tested. Despite the lack of results the research has provided insight into facial expression recognition classification and shown that image processing techniques can be useful.

Gaurav (2018) [8] wrote a case study that discussed real-time facial expression recognition with deep learning. The case study outlined the objectives and constraints associated with facial expression recognition. Accuracy is a constraint in deep learning models and the higher the accuracy the better it will perform in the application and the less errors are made. The case study also outlined three performance metrics that can be used in the evaluation of the deep learning model. These are the *Multi-Class Log-loss*, *Accuracy*, and *Confusion Matrix*. These performance metrics will be discussed on their own in another section of the research document. The model was trained on both human faces and faces of animated characters with an approximate ratio of 1:9. The cross-validation accuracy on animated characters was 100% and 87% on human images. The VGG-16 convolutional neural network was used, this architecture will be discussed in detail in another section of the report. Three human face expression datasets were used as some datasets were not large enough. In total there was approximately 1,500 human images and 9,000 animated images.

The below confusion matrix displays the results tested on the human data. The predicted class is plotted on the x-axis and the true values are plotted on the y-axis. E.g. The “Angry” class was predicted 25% of the time when the actual class was “Disgust”. The model predicted “Angry” when presented with negative face expressions such as disgust and sadness.

In conclusion this case study has provided a lot of value to this project as it introduced many aspects of deep learning which includes the datasets, training, validating and testing the model, performance metrics for evaluating the model and the neural network architecture. These served as a starting point for further research. This case study also proved that animated images may be used to help create a facial expression recognition model that can classify facial expressions of real people.

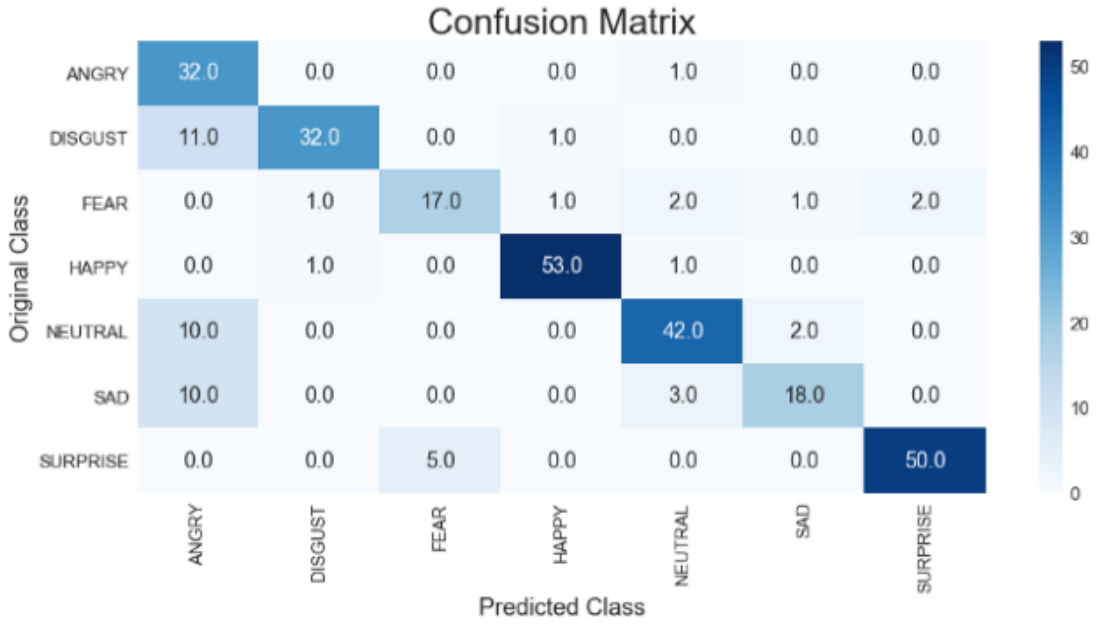


Figure 5

2.1.3 Usability Testing Research

Bastien (2009) [7] reviewed the methods and procedures in usability testing. The usability testing applications the research is based on are in the field of medical and health care informatics. *UsabCheck* will be designed to allow usability testing on virtually all desktop applications, meaning this research is relevant. They discussed the research on remote usability testing. Usability testing in a lot of cases involves the researcher/evaluator inviting the participant to their research facility and the participant completing the tasks in a test room. This room would contain the equipment to record the participant completing the tasks. For example, the recordings can be visual or/and audio. In remote usability testing the participant and the researcher are not in the same location.

Remote usability evaluation can be synchronous, meaning that researcher can manage the usability test in real-time and the interaction between researcher and test participants can be facilitated with conferencing applications. Usability evaluation can also be asynchronous, meaning that the observers are not present during the test and cannot interact with the participant as they are completing the tasks. The advantage of remote asynchronous evaluation is many usability tests to be conducted at one time in parallel and recordings and data collected can be evaluated at another time. For example, the participant will download software onto their machine that will give them instructions for conducting the usability test.

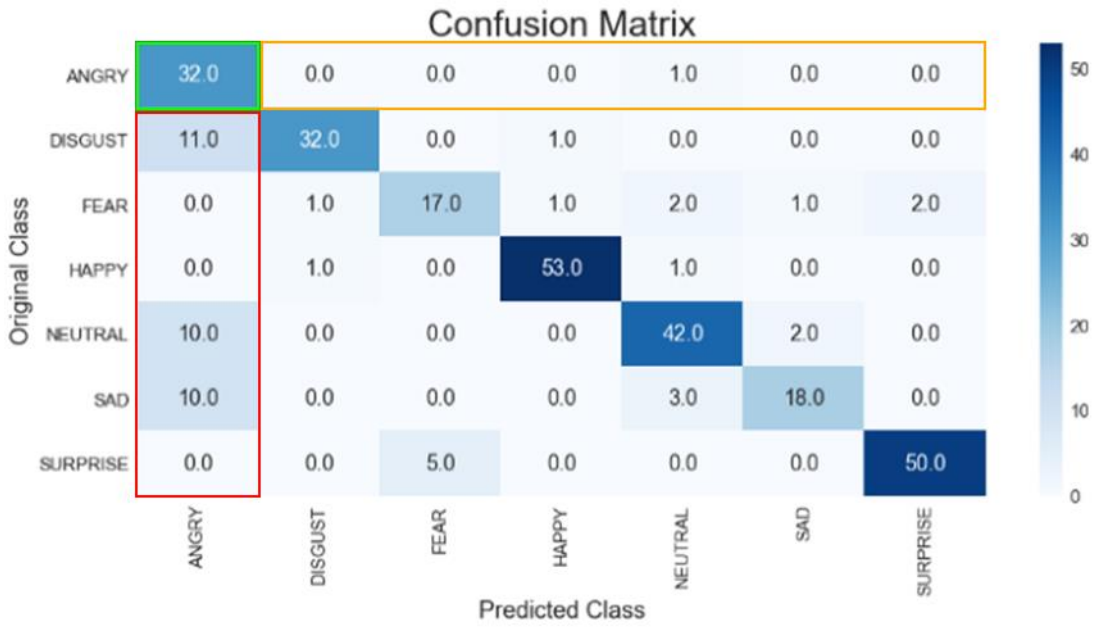
The disadvantage is that the user's equipment such as the microphone and camera need to be of a certain standard to collect the data. Variables such as lighting conditions can reduce the accuracy of the facial expression recognition algorithm.

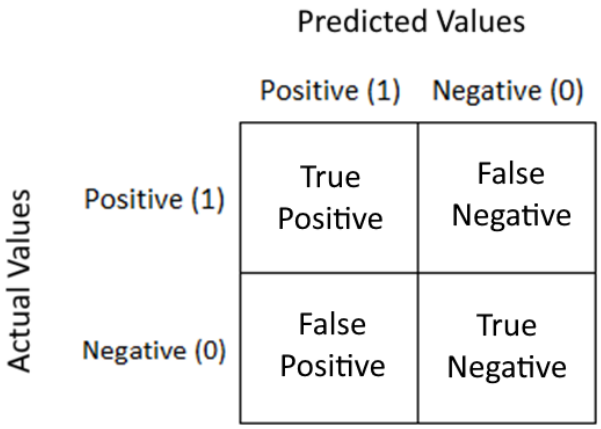
This research has provided valuable insight into conducting usability tests and raised questions to be discussed when designing the *UsabCheck* application with regards to remote testing.

2.1.4 Machine Learning Research

The article by Narkhede (2018) [9] explains the confusion matrix and how the values can be used to calculate precision, recall, accuracy, specificity and the AUC-ROC curve. The cell where the row and column with the same label match is the "True Positive" value. In other words, the accuracy of the model guessing correctly. The "True Negative" value is when the model has predicted correctly that the result is negative. The "False Negative" value is when the model wrongly predicted negative when it should have predicted positive. The cells in the same row show the "False Positives" where the model wrongly predicted a value as positive when it should have predicted a value as negative.

To put the theory into context of emotion classification we look at "Angry" column in Figure 7. Outlined in green is the correct predictions made by the model. The x-axis label and y-axis label for that cell are both "Angry". The cells within the column that are outlined in red and shows the number of times "Angry" has been predicted when the actual label was not "Angry", for example, "Disgust". The cells outlined in orange show the times when the model wrongly classified "Angry" as another class. E.g. The emotions was classified as "Neutral" when it was "Angry".





The article by Koehrsen (2018) [10] explains in detail the classification metrics of recall, precision, accuracy and the F-measure. Accuracy on its own is not a good enough metric to determine if the model is useful if the sample size is large and the ratio of the classes is extremely uneven. This is known as the imbalance classification problem. Increasing the recall means decreasing the precision and vice versa. Depending on the type of application, the design decision might be to increase precision over recall or recall over precision and this will be considered in the design of the *UsabCheck* application.

The *Recall* is the ability of the model to correctly predict.

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

The *Precision* calculates how many positive classes were actually positive.

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP}$$

The *Accuracy* is how much out of all the classes have been predicted correctly.
The *F-measure* combined the recall and precision values to more easily compare two models with high recall and low precision and vice versa.

$$\textbf{\textit{F - measure}} = \frac{2*Recall*Precision}{Recall + Precision}$$

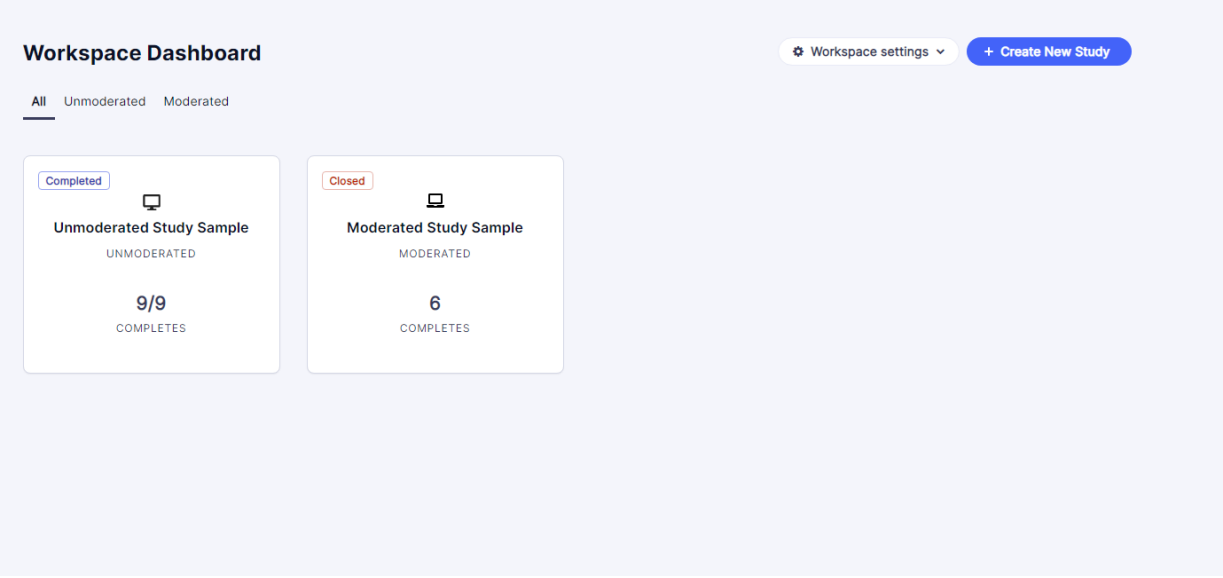
2.2. Alternative Existing Solutions to Your Problem

2.2.1 UserZoom Go

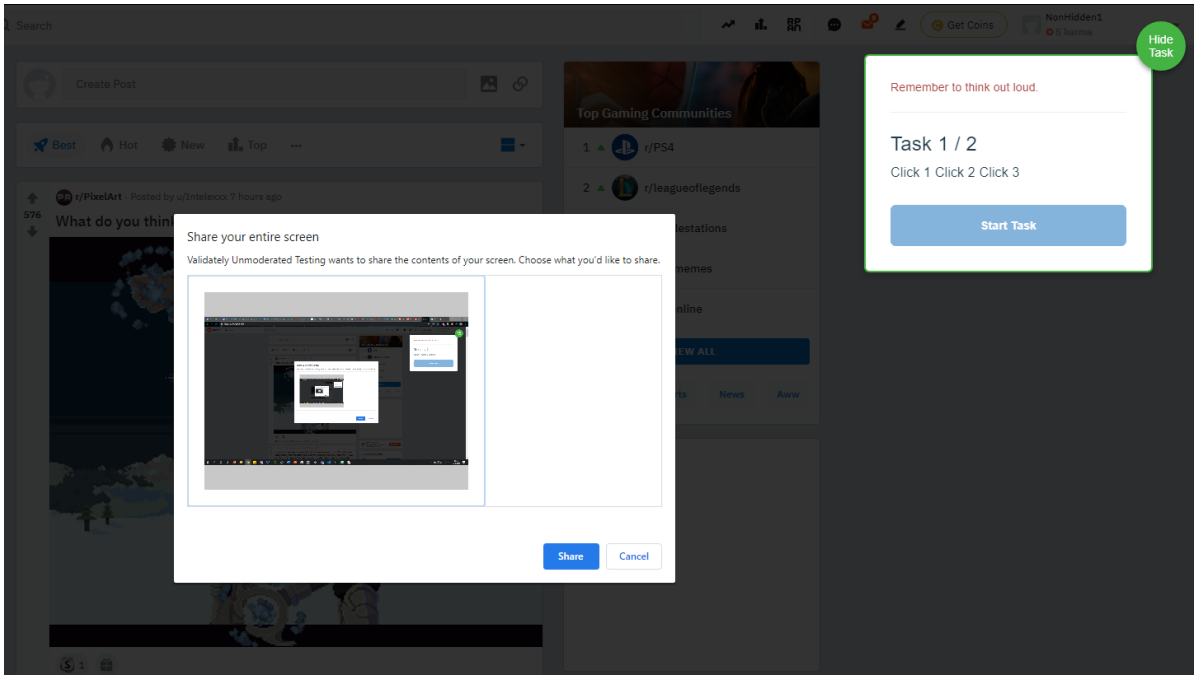
UserZoom Go [11] is a website usability testing application in the browser. It has a clean and easy to understand user interface. The researcher can create a "study" which is a usability test. There is an option to create an unmoderated or a moderated study and these studies can be conducted on own users or paid users can be selected based on a desired demographic based on age, gender etc. The application also provides support for both desktop and mobile applications.

The researcher can then create tasks and questions for the user to fill out and complete. These question answers include plain text, a multiple-choice selection and a rating selection. Questions may be asked before, during or after the study. Once the user begins the test, tasks will be displayed in a small window on screen and the user will click the "Complete" button whenever they are done with the task and move on to the next one. The user's screen, audio and camera may be recorded depending on the usability test specifications. Once the test is over the data will be uploaded automatically and the researcher can view the recordings. The tasks are then reviewed and graded by the researcher with either a "Pass" or a "Fail".

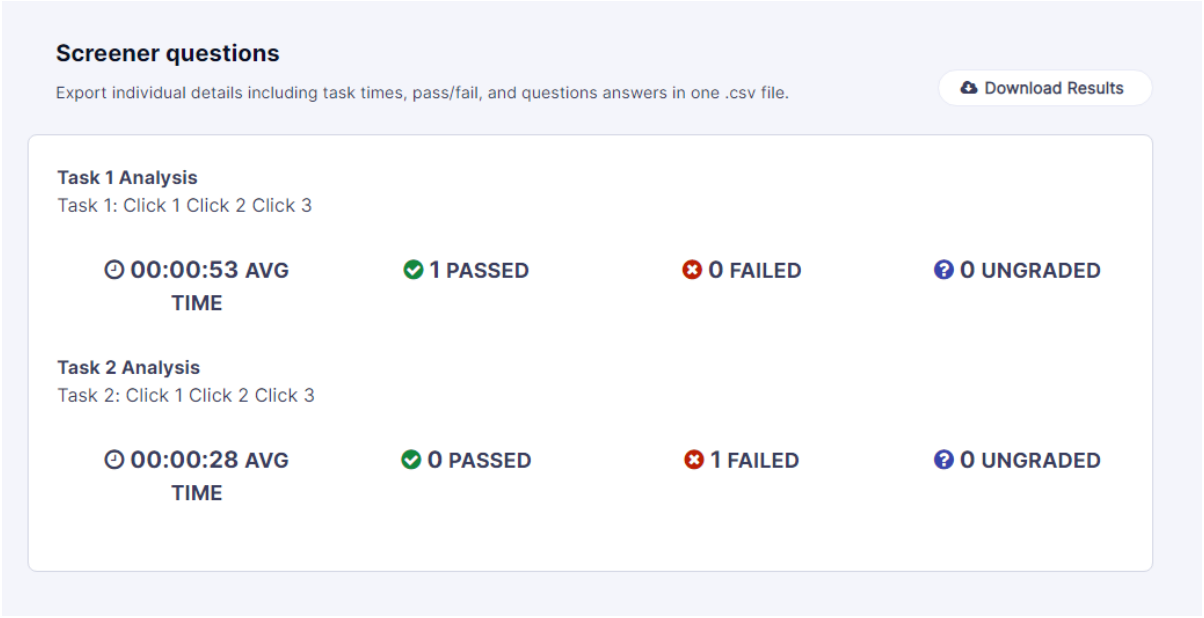
In conclusion, the website is very well designed and implemented. However, this comes at a starting cost of $250 a month for a basic plan which allows only 1 researcher seat and 15 studies per year. The displaying of data could be greatly improved upon as there is no charts that intuitively display the data and no averaging of data is done to provide a general overview. Despite these shortcomings the application is visually appealing and provides the features that make it complete in many aspects.



Dashboard



Usability Test Start



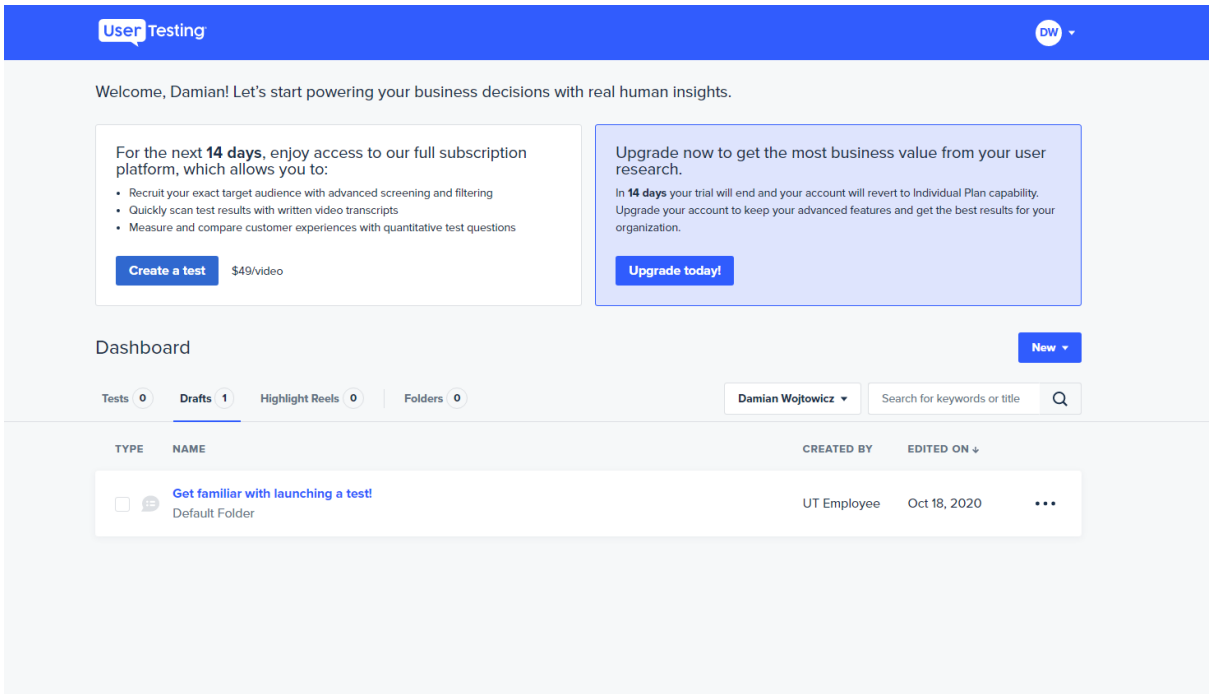
Usability Study Results

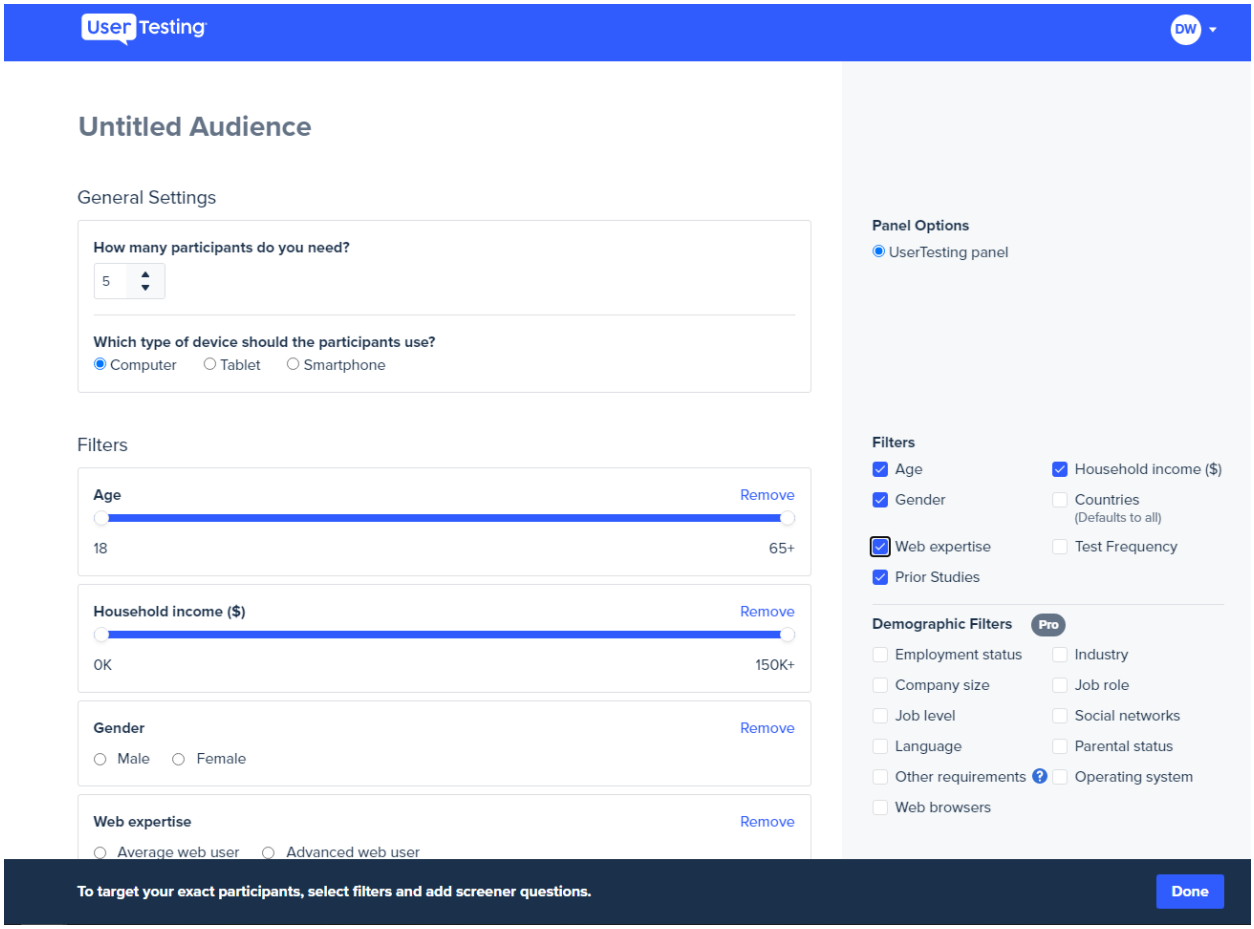
2.2.2 UserTesting

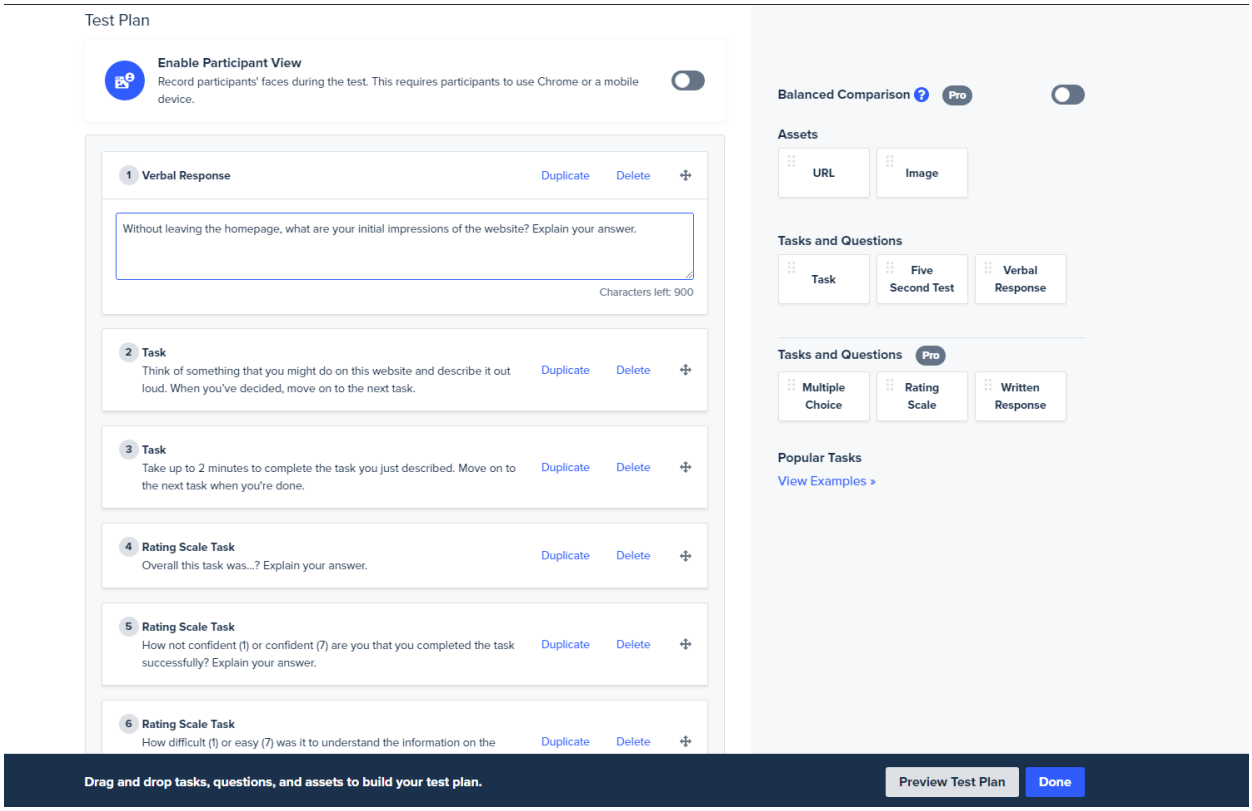
UserTesting [12] is a website usability testing application in the browser. The application that records the screen and the audio has to be downloaded. The user interface was a somewhat disorientating at first as there was a draft test that was already predefined and the button to create a test had a price tag of $49 next to the button. The steps to be taken from the dashboard were not obvious as the initial thought process was to create a usability test and launch it. However, the website suggested launching a draft test and creating a new test appeared to cost money.

Creation of a usability test for free was not an option, however for research purposes the test creation steps were taken before the payment. The test options included selecting the audience by demographic such as age, gender, income, country, employee status etc. The participant device selection was between a computer, a tablet or a smartphone. A scenario could be written for the participant before the test begins and pre-test questions could also be defined. The tasks are written in text and a 5 second task can be set up where a user will be shown a screen for 5 seconds and asked to describe what they saw. A verbal response question can be asked and some question options such as a multiple-choice question, a rating scale and a written response require a premium subscription. There is an option to notify team members of the test results via Slack or email.

The website provided a usability test dry run from the participants point of view before a usability test is launched. This provided more insight into usability testing; however, the data was not recorded as it was simply a dry run and seeing how the data is presented after a participant has completed a test was not possible without payment. In conclusion, the usability testing website provided a lot of options, some of which seemed unnecessary. For example, filtering the demographic by “Parental status”. Viewing the results of a test was impossible without payment despite the free trial. Certain parts of the website were disorientating.




UserTesting
DW
Untitled Audience
General Settings
How many participants do you need?
5
Which type of device should the participants use?
Computer
Tablet
Smartphone
Filters
Age
Remove
18
65+
Household income ($)
Remove
0K
150K+
Gender
Remove
Male
Female
Web expertise
Remove
Average web user
Advanced web user
Panel Options
UserTesting panel
Filters
Age
Household income ($)
Gender
Countries
(Defaults to all)
Web expertise
Test Frequency
Prior Studies
Demographic Filters
Pro
Employment status
Industry
Company size
Job role
Job level
Social networks
Language
Parental status
Other requirements
Operating system
Web browsers
To target your exact participants, select filters and add screener questions.
Done


Test Plan
Enable Participant View
Record participants' faces during the test. This requires participants to use Chrome or a mobile device.
1 Verbal Response
Duplicate
Delete
Without leaving the homepage, what are your initial impressions of the website? Explain your answer.
Characters left: 900
2 Task
Think of something that you might do on this website and describe it out loud. When you've decided, move on to the next task.
Duplicate
Delete
3 Task
Take up to 2 minutes to complete the task you just described. Move on to the next task when you're done.
Duplicate
Delete
4 Rating Scale Task
Overall this task was...? Explain your answer.
Duplicate
Delete
5 Rating Scale Task
How not confident (1) or confident (7) are you that you completed the task successfully? Explain your answer.
Duplicate
Delete
6 Rating Scale Task
How difficult (1) or easy (7) was it to understand the information on the
Duplicate
Delete
Balanced Comparison
Pro
Assets
URL
Image
Tasks and Questions
Task
Five Second Test
Verbal Response
Tasks and Questions
Pro
Multiple Choice
Rating Scale
Written Response
Popular Tasks
View Examples »
Drag and drop tasks, questions, and assets to build your test plan.
Preview Test Plan
Done

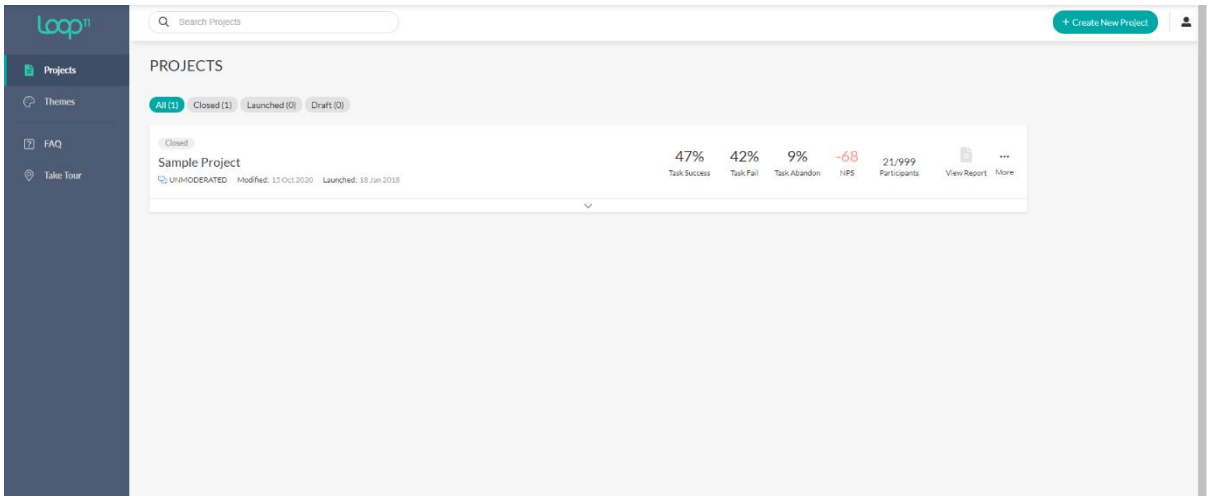
2.2.3 Loop11

Loop11 [13] is a website usability testing application in the browser. The researcher can create a new "project" which is a usability test/study. Statistics and other information about the general performance is given in the project overview and pie charts display the number of participants that have successfully completed the tasks, failed or abandoned the study. Information on the total number of participants and the average duration of the study is displayed. More statistics on the participants is displayed such as their location, operating system, device and browser.

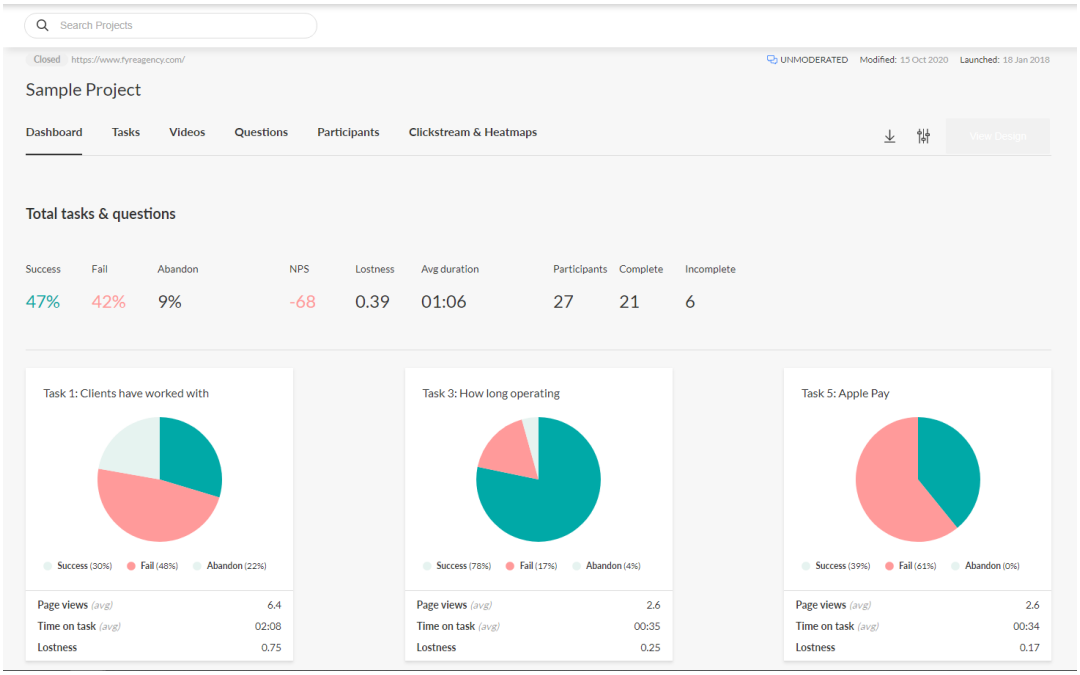
The researcher can view the project in terms of the tasks, the videos, questions and participants. A tab is given for each of the mentioned. This allows for easy navigation which is definitely desirable. The video of the participants completing the tasks and questions can be viewed. All the data on each participant can be displayed. A clicksteam & heatmap can be displayed which shows how users progressed through the tasks in terms of the links they clicked on.

The tests can be moderated or unmoderated and there are options to enable or disable screen and webcam recording for the test. Tasks are written in plain text with the option to customize the text with fonts and font sizes. There is a wide selection of questions that can be selected from ask the participant. These include a multiple-choice question with one answer, a rating scale matrix, a ranking question and a multiple choice (multiple answers) question just to name a few. There are options to randomize the order of answers in a given question when applicable e.g. multiple-choice question.

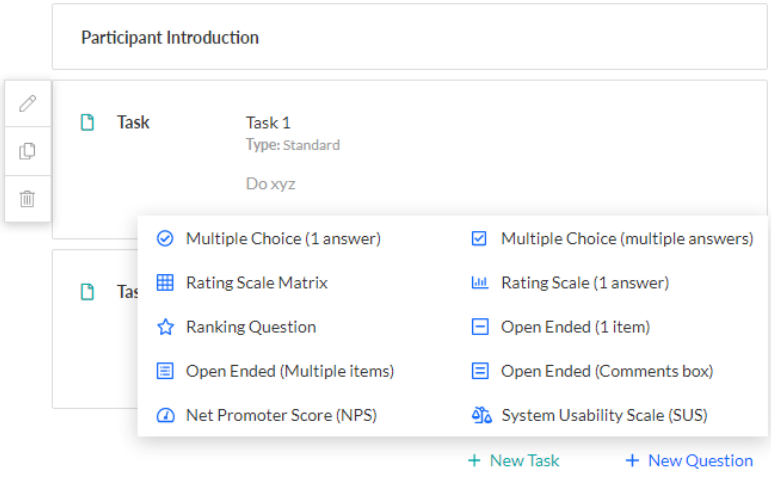
In conclusion, this application has a modern design that is appealing to the eye and easy to use. Navigation is easy and the data is displayed in a presentable manner which includes charts. The clickstream is an interesting idea of tracking how all the users progressed in the task. An issue with the website is that the video quality was very low and would not load very fast.



Main Dashboard

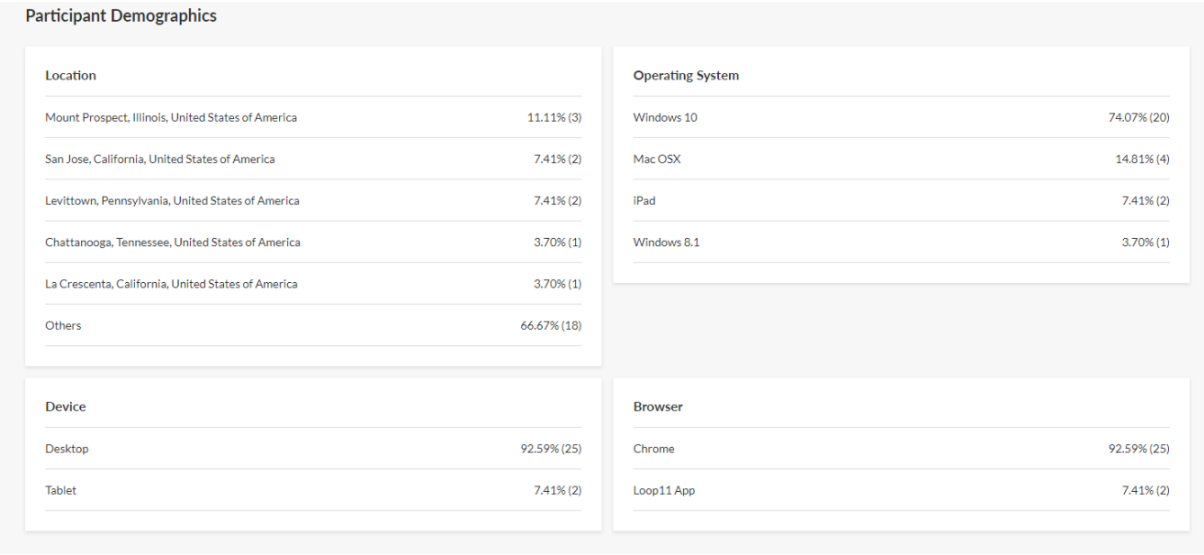


Project Dashboard



Tasks and Questions

Clickstream



Participant Demographic

2.3. Technologies Research

2.3.1 Machine Learning

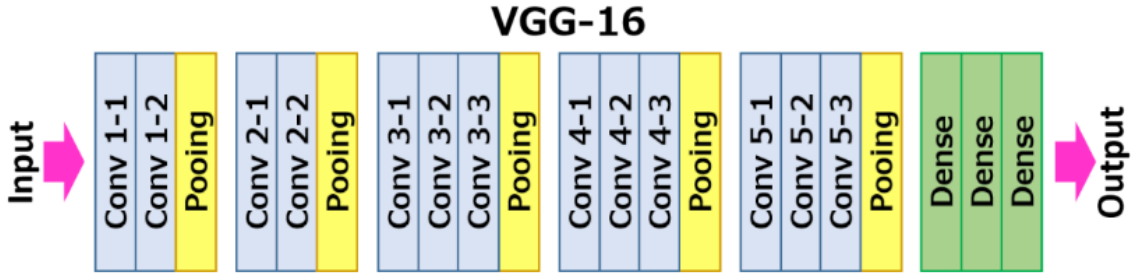
2.3.1.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of Deep Neural Network (DNN) and can efficiently and effectively be used to perform numerous tasks that involve pattern and image recognition. These tasks include object recognition, hand gesture recognition [Citation] and face recognition [Citation], to name a few. CNNs may be used for image segmentation, classification and retrieval with high accuracy. For example, a CNN trained on the MNIST (handwritten digits dataset) has achieved a detection rate of 99.77% [Citation] which shows how effective CNNs in image classification.

A CNN consists of layers and these layers include, the input layer, convolutional layers, pooling layers, rectified linear unit layer and a fully connected layer. The input layer is the first layer that takes in the input data. The convolutional layer applies filters to extract the features from the data and a CNN may involve the use of many convolutional layers. The extracted features are passed to the pooling layer which reduces the size of the images. This preserves the most prominent features which is extracted. The Rectified Linear Unit Layer (ReLU) is an activation function that replaces any negative number in the pooling layer with zero. This keeps the CNN mathematically stable and prevents the vanishing gradient problem when the number of layers increases [Citation]. The fully connected layer is the final layer and uses the results from the previous layer to classify an image [Citation] [Citation].

2.3.1.2 VGG Model

The VGG-16 Model [Citation] is a famous CNN model that achieved a 92.7% test accuracy in the ImageNet dataset [Citation], making it the winner of the ILSVRC (ImageNet Large Scale Visual Recognition Competition) in 2014 [Citation]. The dataset the model was trained on consisted of 14 million labelled images belonging to 1000 classes [Citation]. This model can be used to train a facial expression recognition model as shown by Gede Putra Kusuma *et. al* 2020 [Citation] . They used a fine-tuned VGG-16 model on the FER2013 dataset and achieving an accuracy of 69.40%, outperforming most standalone-based model results.



2.3.1.3 Support Vector Machines (SVM)

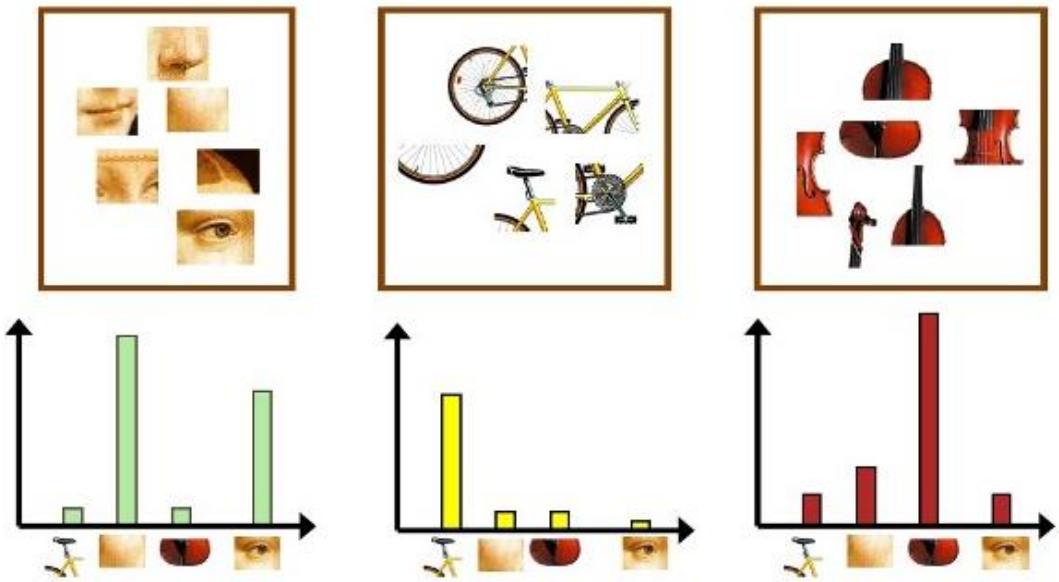
Support Vector Machines (SVMs) can be used for both regression and classification tasks. The algorithm works by finding a hyperplane in an N-Dimensional space where N represents the number of features. The hyperplanes separate the data points and the optimal hyperplane has a maximum margin between the data points of the classes. The accuracy of the algorithm depends on the hyperplane to separate the classes. In other words, the smaller the margin the more difficult it is to classify the data points as belonging more to one class than the others [Citation].

In the scenario that the data points are not linearly separable, the SVM kernel trick can be used. The concept behind the kernel trick is to map the dataset into a higher dimensional space which enables the algorithm to find a hyperplane that can separate the data [Citation] [Citation].

2.3.1.4 Bag of Visual Words (BOVW)

Bag of Visual Words is commonly used in image classification. The concept is similar to the Bag of Words (BOW) in natural language processing. In BOW the frequency of the words is obtained and can be displayed on a histogram. In BOWV, instead of counting the frequency of words in a given text, the image features are counted and can be displayed in a similar fashion using a histogram. Image features consist of a keypoint and a descriptor. Keypoints are segments of the image that stand out and this can be determined by the corners and edges in the image. These keypoints do not change even if the image is resized or rotated. The descriptor is the description of the keypoint.

The descriptors are clustered with a clustering algorithm. An example of a clustering algorithm is K-Means clustering and the center of each cluster is the visual "word". A histogram is generated for each training images and to classify an image the features are first extracted, then plotted on a histogram and compared with the visual words' histogram from the training set [Citation] [Citation].



2.3.2 Dataset Research

2.3.2.1 JAFFE (The Japanese Female Facial Expression (JAFFE)

The JAFFE dataset consist of labelled images of the head. The dataset consists of 10 people, all of whom are Japanese and female. There is a total of 213 grayscale 8-bit images and the image resolution is 256x256 pixels. The facial expressions are labelled with anger, disgust, fear happy, sad, surprise, and neutral. For a high accuracy model, a total of 213 images is not sufficient and data augmentation techniques will have to be used to generate more images from the existing set to increase the amount of data.

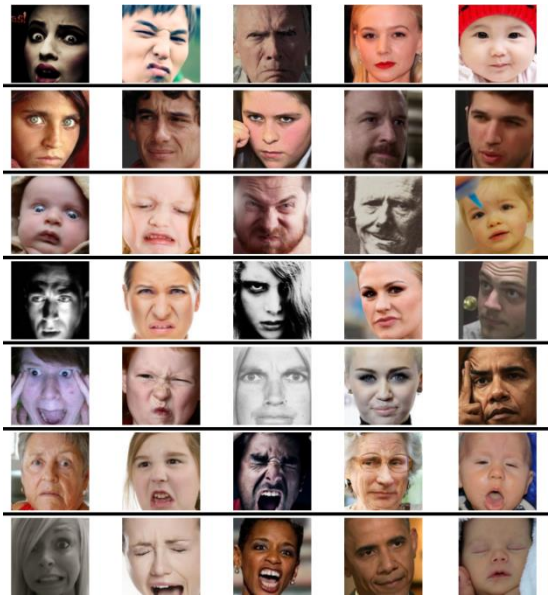
2.3.2.2 FER2013 (Facial Expression Recognition Dataset 2013)

The FER2013 data is in the form of a .CSV file and consists of 28,000 labelled images in the training set, 3,500 images in the development set and 3,500 images in the test set. The images are stored in the file as an array of pixel values. These can be used to reconstruct the image to a .jpg or .png format. The images are labelled with one of seven emotions which are happy, sad, angry, afraid, surprise, disgust and neutral. All images are grayscale and of 48x48 pixel resolution.

2.3.2.3 AffectNet

AffectNet is one of the largest datasets for facial expression recognition and consists of more than a 1 million images with approximately 440,000 of which are manually annotated. The publishers of the dataset are experiencing issues with bandwidth as a result of many downloads meaning the request for the dataset is delayed and a response is yet to be received. The sample images appear to be of the face only and the very large quantity of images would increase the accuracy of the FER model.

Neutral	75374
Happy	134915
Sad	25959
Surprise	14590
Fear	6878
Disgust	4303
Anger	25382
Contempt	4250
None	33588
Uncertain	12145
Non-Face	82915
Total	420299



2.3.2.4 FacesDB

The FacesDB dataset consists of 38 subjects who are both male and female and each face expression is associated with a subject such as joy, sadness, fear, anger, disgust, surprise and a neutral face expression. This means that there are only 38 images for each face expression and these images are of the frontal point of view with side views of the face explicitly labelled. Due to the quantity of images the dataset images will have to be augmented to generate more.

2.3.2.5 FEC (Goole Facial Expression)

The FEC (Google Facial Expression Comparison dataset) is in the form of a .CSV file. The dataset consists of face image triplets with each row in the file contains three faces. It is specified which of the two images in the triplet contain the most similar face expression. The dataset was rated and annotated by at least six people. The dataset itself consists of approximately 500,000 triplets and approximately 156,000 face images. The images can involve people doing activities which is unlike some of the other datasets which only include the head. In the CSV file the image, four coordinates are provided that form a bounding rectangle that locates the face.

As the dataset is in the form of a .CSV file the images have be downloaded and cropped using the bounding rectangle coordinates. The process by hand is simply unfeasible and a GitHub repository under the name “FEC Dataset Downloader” was discovered for an application [14] that was said to download all the images from this particular dataset automatically. However, that application used another application under the name “TorCrawler” [15] that was in nature of a web crawler and utilized the TOR browser to rotate the IP address. The TorCrawler functionality was said to be necessary as the dataset image downloads apparently stall after a while if the IP address is not rotated. The TOR Crawler application included instructions that assumed the operating system used was UNIX which meant the instructions could not be followed as the operating system used is Windows 10.

A decision was made to write a program that will attempt to simply pull down the images without the changing of IP addresses which has been successful. However, after further inspection the images may not be entirely suitable to the usability testing scenario as the images of the faces are taken at many different angles which may reduce the accuracy of the model. In a usability testing scenario, the environment is not going to be “natural” in that the scene is not expected to constantly shift and change as it would if the person was in a bustling area. The camera will be in a static position and the angle of the participant in question is unlikely to greatly vary.

Example of FEC Dataset Image Taken at Angle

2.3.2.6 Emotic (Emotions in Context)

The Emotic dataset is similar in nature to the FEC Google dataset except that is exclusively focuses on images of people in real environments. The images are annotated from a list of 26 discrete categories such as peace, sympathy, anger and surprise to name a few. The annotation are not simply for the face but also include the body. The angle at which the images are taken at and the fact that the focus is not only on the head might pose problems if used to train a facial expression recognition model. This dataset, as the name suggests, focuses on the context which is not a concern in a controlled usability testing environment.

Emotic Sample Image

2.3.2.7 Conclusion

In conclusion, several facial expression recognition datasets have been researched and analysed. Certain datasets such as the FEC dataset and the Emotic dataset were not applicable in the usability testing environment. Other datasets such as JAFFE, FER2013 and FacesDB datasets were perfect for the training of the model in terms of the types of images. However, the JAFFE dataset and FacesDB dataset will require image augmentation to increase the dataset size. The AffectNet dataset is perfect in terms of both the types of images and the amount of images. However, as a result of these desirable features, the dataset is in high demand and the researchers who have published it are struggling with download bandwidth, meaning I could not acquire the dataset.

2.3.3 Programming Languages

2.3.3.1 Python

Python [Citation] is a high-level, interpreted, object-orientated and general-purpose programming language. Python supports a vast amount of libraries which includes machine learning libraries such as TensorFlow, Keras and NumPy, to name a few. The language is free, open source and cross platform. The programming language is popular and finding support online to questions is easy which greatly helps with the development process. The popularity of the language with regards to machine learning and data visualization makes it a very strong contender as a language of choice for machine learning. [Citation]

Django [Citation] is an example of a high-level Python Web framework which allows for web development with Python. Django is integrated with JavaScript and HTML which means JavaScript code and functionality can be used in Django.

2.3.3.2 Java

Java [Citation] is a class-based, object-oriented, platform-independent language. Java can be used to develop the server side of web applications. Java is the no. 1 choice for developers with over 9 million Java developers worldwide [Citation]. The popularity of the language and its use in web development makes it an excellent option for developing the UsabCheck web application. It is a programming language taught in the TU Dublin Computer Science course which makes it familiar and easier to work with.

2.3.3.3 JavaScript

JavaScript [Citation] is an interpreted programming language that allows developers to implement complex and dynamic features on web pages. JavaScript compiles the code at run time with the use of a technique called *just-in-time compiling*. JavaScript can run locally on the machine of the user viewing the website and can be used for interaction with the back-end server via a Restful API. It can also run on a server. There is a back-end and front-end framework named Node.js [Citation] and there are front-end frameworks such as AngularJS [Citation] and ReactJS [Citation] if the design plan will be to have a Java back-end as an example.

2.3.4 Python Libraries

NumPy [Citation] is an open source library used for working with numbers, arrays, matrices, linear algebra etc. Images are simply matrices of pixel values and NumPy can be used to alter and manipulate these images.

Pandas (Python Data Analysis) library [Citation] provides functionality that can extract data out of a CSV file as an example and create data frames that are similar in structure to an Excel file. This library can work with NumPy to manipulate the data.

Matplotlib [Citation] is a library that generates charts and other forms of data visualization in Python. These figures can be interactive and allow to user to zoom in and pan etc. This library can be used to display images in Juypter Notebook and the results of machine learning training in the forms of a confusion matrix as an example.

Scikit-learn [Citation] is a machine learning library that includes other libraries such as NumPy, SciPy and Matplotlib. It is a tool for predictive data analysis and includes implementation of algorithms such as SVM, Nearest Neighbour, Random Forest, K-Means etc.

TensorFlow [Citation] makes it easy for developers to create and train deep neural network models for desktop, mobile, web etc. This library has many applications which include sound recognition, text-based applications, image recognition and video detection [Citation] .

Keras API [Citation] is fully integrated with TensorFlow and serves as a wrapper for TensorFlow and Theano, providing a simple and intuitive interface for the machine learning libraries.

FastAI [Citation] is another deep learning library that provides its users with high-level components and it is GPU optimised just as the other libraries mentioned. An article [Citation] detailed the first impressions of the FastAI library and impression was that it is a good library to use however it comes with some slight learning curve drawbacks. The wiser decision would be to use TensorFlow and Keras to train the deep learning model as the two libraries are more popular and there is more online support as a result.

OpenCV [Citation] is a computer vision, machine learning and image processing library. It can be used to manipulate images which includes resizing, rotating, detecting edges, converting the image to different colour spaces etc. This library can be used in a hybrid approached whereby features are extracted with OpenCV and used to train a machine learning algorithm.

2.4. Usability Testing Technologies

2.4.1 Usability Testing

As previously explained the idea for the usability testing is to create a web application that will store the data and allow researchers to configure the usability tests. For this goal to be achieved video uploading and streaming has to be implemented. The website has to be hosted and a database will have to be selected.

2.4.2 Video Uploading

Several video streaming services have been researched to find the most convenient method of uploading and streaming the videos reliably.

2.4.2.1 SwiftStack

SwiftStack [Citation] is a data storage and management platform for data-intensive applications. It is easily scalable and supports HTTP Range requests which means that pseudo-streaming is supported. Pseudo-streaming involves downloading the file and playing that file as it is being downloaded. This is how YouTube streams their videos.

2.4.2.2 StreamingVideoProvider

StreamingVideoProvider (SVP) [Citation] provides video streaming and other services. The videos uploaded can be embedded in the *UsabCheck* application. SVP provides video tutorials for how to use their services and a developer's API for cURL, Java, PHP, Ruby, Python and JavaScript. The videos can be managed and statistics can be provided. The service provides more functionality than is needed for the *UsabCheck* application. However, this is not a problem.

2.4.2.3 Vimeo

Vimeo [Citation] is an online video streaming site that allows users to upload their videos and these videos can be shared and embedded on websites. Videos can also be uploaded to Vimeo much the same as YouTube.

2.4.2.4 YouTube

YouTube is a video streaming service and videos can be uploaded and the videos can be configured in such a way that only people with the link may view the video. YouTube also provides an API that allows users to upload the video. However, YouTube sets a restriction on videos uploaded via an API and only a few videos can be uploaded each day. To increase the amount of videos that can be uploaded with the API a quota has to be requested.

2.4.3 Website Hosting

2.4.3.1 Tomcat

Apache Tomcat [Citation] is a lightweight application server designed to execute Java servlets and render web pages that use Java Server page coding. It provides a relatively quick load and redeploy times. Apache Tomcat is most widely used Java application server and it is well documented as it has been around for almost 20 years [Citation].

2.4.3.2 Google Firebase

Google Firebase [Citation] is a Backend-as-a-Service application development software that enables developers to develop iOS, Android and Web apps. It is the server, the API and the datastore all in one. However, the database, the Firebase Realtime Database and Firestore are non-relational databases. This means that storing and retrieving data is stored in documents and collections unlike a SQL database. This makes is very fast and efficient, however, not all applications are suited to this type of storage and this will be further discussed in the design section.

2.5. Existing Final Year Projects

Existing projects which involved usability testing have not been found. However, there have been many projects that have used machine learning. One such example is a project by Brendan Tierney, titled “Detecting Bot Twitter Accounts using Machine Learning”. The purpose of the project was to use machine learning to evaluate whether or not a Twitter account is a bot. The complexity of the project stemmed from the machine learning aspect which involved the use of multiple machine learning models as a way to more accurately a bot account. There was a model for classifying based on the user data, the tweets of the user, the sentiment and timing. Another form of complexity came from testing various machine learning algorithms to see which one performs best. The project also faced challenges such as the accuracy of the dataset and the speed of the application.

The project technologies included the Django framework, JavaScript, Bootstrap, HTML, and CSS. MySQL database and the Twitter API. Vast amounts of research have been done into each area. The challenges and problems were identified, and a solution was proposed. The project implements features that the existing applications lack which was detecting the followers of an account. The weakness of the project was that the follows of a user were not processed and identified fast enough in some cases because of the constraint imposed by the Twitter API.

2.6. Conclusions

Conclusion will be done later as the entire project may need to be restructured as a conclusion has been provided for each individual section.

3. Prototype Design

3.1 Introduction

In this section the software methodology will be discussed, and the system design will be discussed in detail. The design of the system will include the front-end, back-end and database of the web application and the local python application with the facial expression recognition model.

3.2. Software Methodology

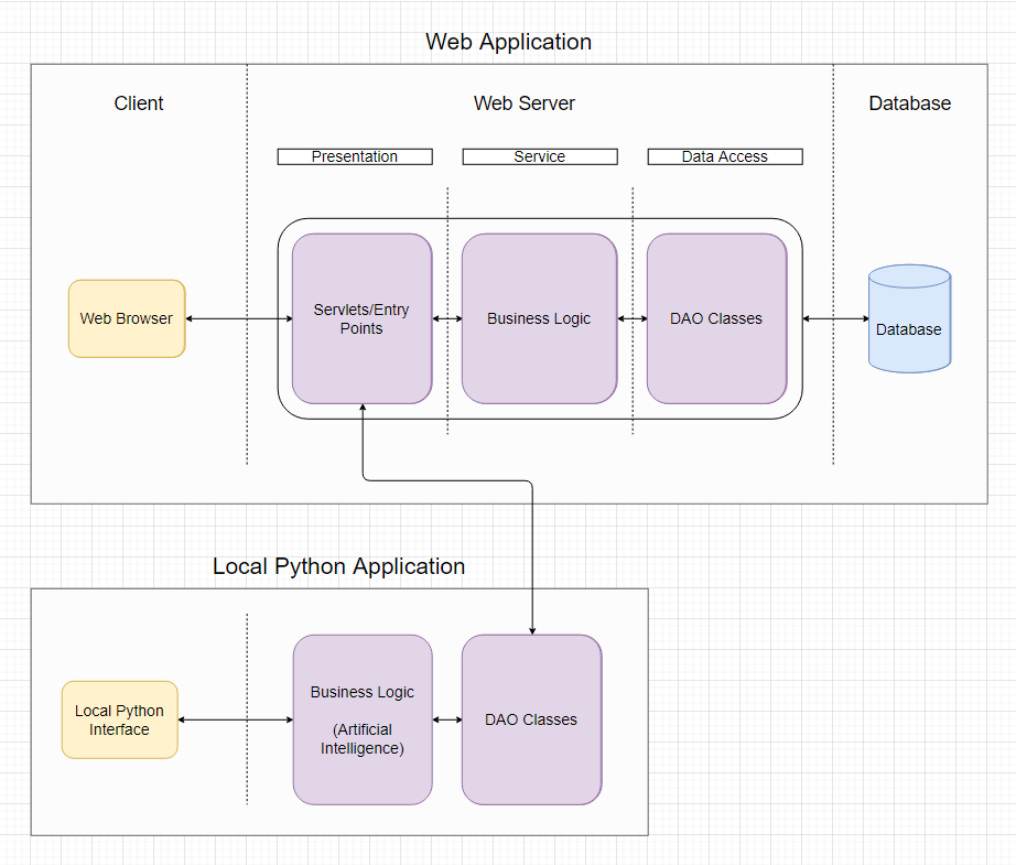
The software methodology that has been chosen for this project was the Scrum Agile methodology [Citation]. The Agile methodology is centered around iterative development and allows teams to deliver work with greater predictability and allows for better adaptability to change. In Scrum specifically, the development is split into development cycles called Sprints which are at least no more than four weeks. The project's high-level requirements and goals are defined in the Product Backlog is a dynamic list of Product Backlog Items (BPIs). This Product Backlog can change depending on the requirements as the project develops. These BPIs are used as input to the Sprint Backlog, which is a list of tasks, user stories, bug fixes etc. which need to be completed by the end of the sprint. At the end of each sprint the team reviews and evaluates the work. The Scrum methodology also involves "Increments" which is the usable end-product from a Sprint [Citation].

Although this methodology is most commonly applied in a scenario with a team, the methodology itself provides a very useful process for managing a project with only a single individual. The product quality is improved as a result of breaking down the project requirements into manageable pieces and the focus is on the user with the project being able to adapt to the changing requirements. Prioritizing and reviewing the tasks increases the productivity of the developer as they can easily track the development and readjust if necessary [Citation].

Diagrams will need to be added

3.3. Overview of System

This project involves two applications. One application runs locally on the participants computer where the usability test will be conducted. Another application is the web application which involves the front-end, back-end and database.



4. Prototype Development

As least 2 pages, but as many as you like (but lots of code samples).

4.1. Introduction

4.2. Prototype Development

4.3. Front-End

4.4. Middle-Tier

4.5. Back-End

4.6. Conclusions

5. Testing and Evaluation

As least 2 pages, but as many as you like

5.1. Introduction

5.2. Plan for Testing

5.3. Plan for Evaluation

5.4. Conclusions

6. Issues and Future Work

As least 5 pages, but as many as you like

6.1. Introduction

6.2. Issues and Risks

6.3. Plans and Future Work

6.3.1. GANTT Chart

Bibliography