Final Project Report

Prepared by Group: **FIT3162_MA_8**

Project Supervisor: **Muhammad Fermi Pasha**

Word Count: 8,763

AI-assisted Online Psychotherapy Portal

22 October 2021

<u>Group Members:</u>

Grace Nathania (30241510)
Gao Hui Min (30212928)
Mohit Hotchandani (30206235)
Goh Ngee Juay (30236681)

TABLE OF CONTENTS

1) Introduction

The aim of our project is to streamline the management and administration process behind the online psychotherapy services. Psychotherapists, which provide online services, would often find themselves spending a bulk of time on administrative tasks: keeping records of patients, compiling data and tagging data. These administrative tasks distract psychotherapists on what they do best, which is providing counselling services to patients. With the aid of our software, we hope to ease the burden of manual work by automating some of the tedious and routine tasks which include storing recorded counselling sessions, keeping track of patient records, and storing metadata of videos.

Psychotherapists might need to look back on the recorded counselling sessions to identify the periods where patients show a strong emotional response to verbal questions and prompts. These might be the important periods of the counselling process where psychotherapists would take note for further analysis. However, they usually would have to search the videos manually for these detected emotions. With the aid of our software, the AI will automatically label each time segment where an obvious emotion occurs, thus relieving the tedious work of going through a long recorded counselling session such as a thirty minute video.

There are two main components to the project: web application and deep learning networks. The first component is the web application which serves as a portal where users (i.e. psychotherapists) can access different functionalities. Users can manage and update their patient's profiles, upload videos of counselling sessions, and perform searches on patient's details and metadata. All of these generated data are stored in the database and seamlessly displayed in the relevant pages for the user. This reduces the cognitive load of users and chance of human error when it comes to dealing with the tasks of managing the data.

Secondly, the web application has an AI component which employs machine learning to automatically tag videos based on recognized emotions. This will enable the users to skip to the relevant timestamps where emotions are detected. With the help of the machine learning backend, users can focus on the relevant periods of the videos that might be interesting to them. The target audience of this software is mainly child psychotherapists who provide online psychotherapy sessions. In the regards of the target audience, the Facial Expression Recognition (FER) deep learning (DL) model is trained on children's images and it works best for patients in that child age (< 18 years old) category. The use of the DL model represents the AI part of the psychotherapy portal.

The project we have implemented is following our initial plan in semester one. We have focused on delivering the project deliverables and meeting the product user acceptance criterias stated in the Requirements Traceability Matrix. Not only that, proper project management has been followed by adhering to the small scale scrum development methodology. Risks are managed according to proper risk management using the risk register. Quality assurance and monitoring is done by performing testing on the functionalities implemented and time management is conducted by reviewing the team's progress through weekly minutes and reports to the stakeholders.

This report attempts to summarise our entire project from its aims and objectives, the methodologies used, outcomes, and a thorough evaluation of the successes and shortcomings of the project. Reasoning and justification for each major decision related to any aspect of the project is also discussed. This is done in a hope to perform introspection and scrutinise the things that made the project a success and where might improvement can be made.

2) Literature Review

This section will briefly summarise the relevant literature review done in semester one. We will also explore which literature has guided our implementation in semester two. The purpose of this literature review is to establish the foundation of knowledge which is used to realize our project. This is especially important for the Deep Learning component of the project where there exists many alternatives to implement the neural network.

2a) Summary of literature review in semester one

In the literature review in semester one, we have found out that online psychotherapy has no statistically significant drawbacks compared to face-to-face psychotherapy interventions as found in Barak et al (2008). This signifies that development of online psychotherapy platforms and services is a worthwhile endeavour to pursue especially given the expansion of online services during the Covid-19 pandemic era.

We have also reviewed literature which surveyed the development of implementation and application of Facial Expression Recognition (FER), namely Li & Deng (2020). This paper serves as an entry point into the implementation for the FER by giving us a broad overview of possible implementations. It also suggested possible flaws in the training of neural networks done in the past. Specifically the lack of quality training data leads to overfitting and low test accuracy. This particular drawback will be a focal point in which many researchers try to overcome.

This view is also backed by Khan et al (2019), which states that this lack of quality data is an even bigger problem in the development of children FER. There is a scarcity of publicly available datasets which can be used and most of them contain posed and artificial expressions with poor real life validity. Therefore, Khan et al has invented a novel emotional database called LIRIS-CSE which attempts to solve the aforementioned limitation. The database contains 208 movie clips with a total of 26 thousand images of children with natural expressions.

This database is used by Qayyum & Razzak (2021) which employs a progressive light residual learning for classification. In semester one, this paper was chosen to be the main source of inspiration for the Deep Learning FER deliverable of the project.

2b) Literature review in semester two

In semester two, we have explored the Qayyum & Razzak (2021) journal article in greater detail. The LIRIS-CSE database used is very related to the project since the users will upload videos of their patient's counselling sessions. It is also a credible source of data since the database contains children expressions which are our primary target patients. However after further inspection, Qayyum & Razzak (2021) article made no mention of how the videos in the database were used to train their DL architecture. Despite that, based on the inputs to their DL network, it was plausible that the researchers sliced the images and fed it one at a time into their network, which essentially treated the database as a source of images rather than videos.

The implication is that this eliminates one of the advantages promised by the new LIRIS-CSE novel database because it contains frame by frame transformation of one expression to another. For example, a sequence of images (i.e. video) may start with a neutral expression in the beginning, and it may gradually transform into a peak of happy expression at full smile. This feature cannot be found in other existing databases which only includes static frames at peak expressions. This is one of the key advantages that this dataset provides. Another advantage promised is that models trained on this dataset should have been more accurate than training on existing child FER datasets. The main reason for this is that LIRIS-CSE contains videos of children reacting to video clips, and thus their expressions are natural and not posed unlike other datasets. We wanted to utilise the sequential and natural aspect of the videos in LIRIS-CSE to the fullest extent and thus the team have made the decision to move away from the plan to use the architecture in Qayyum & Razzak (2021).

Kuo et al (2018) have proposed a frame-to-sequence approach. This allows the model to exploit the temporal information in the data.This is done by taking a sequence of images as input and mapping each of the images to an emotion category. This is accomplished by a conventional Convolutional Neural Network (CNN) which detects an emotion in a static image. To further exploit the sequential nature of the input data, a shallow Recurrent Neural Network (RNN) structure such as GRU can be used to capture the temporal aspect of the sequential input. The progression of the expression is also learnt using this method. In summary, this enables us to produce a more accurate prediction of emotion given a sequence of image data compared to a conventional CNN which takes only a single input image. This is evident from the state-of-the-art accuracies obtained by Kuo et al (2018).

Transfer learning is the process where weights from other pre-trained networks are utilised for other networks with similar objectives (Dilmegani, 2020). According to Dilmegani, major benefits include a better initial model, higher accuracy and quicker training time. Therefore, a pre-trained model can be used as a starting point for another model. A relevant pre-trained model for the project is the Keras-VGGFace2-ResNet50 by Cao et al. This is a Keras model which is trained on a dataset created by the same researchers which consists of around 3.31 million images of more than 9 thousand subjects. By taking advantage of transfer learning, researchers can use the weights and biases learnt by other researchers as a starting point rather than training their network from scratch.

2c) Conclusion

In conclusion, the team has utilized the knowledge obtained from the literature review for the implementation of the project. The following are the main literature that pertains to the project. We have obtained the LIRIS-CSE dataset from Khan et al which is used as the source of children input data for the network. This dataset contains many advantages over pre-existing ones as pointed out by Khan et al. We have modified the architecture of the DL; starting from a light residual network architecture from Qayyum & Razzak to a frame-to-sequence approach from Kuo et al. The specifics of the architecture will be discussed in the methodology section. We have also decided to exploit the advantages of transfer learning by using the VGGFace2 from Cao et al as part of the DL network. This enables us to hasten the training time for the DL as well as the overall development of the project.

3) Methodology

In this methodology section, the final design will be discussed and analysed in comparison to the initial design proposed. The final software specification is also compared with the tentative specification in semester one. Lastly, the activities planned to implement the design will be discussed.

3a) Web application design and software specification

Figure 1 below shows the block diagram that we have planned in semester one for the design of the web application. The web application consists of 4 modules: login/signup, counselling, child and search module. After the development of all the modules of the web application, reflection on the initial design and observation of any changes that took place were taken. In the second part of working on the web application, the initial design has been implemented with some adjustments.

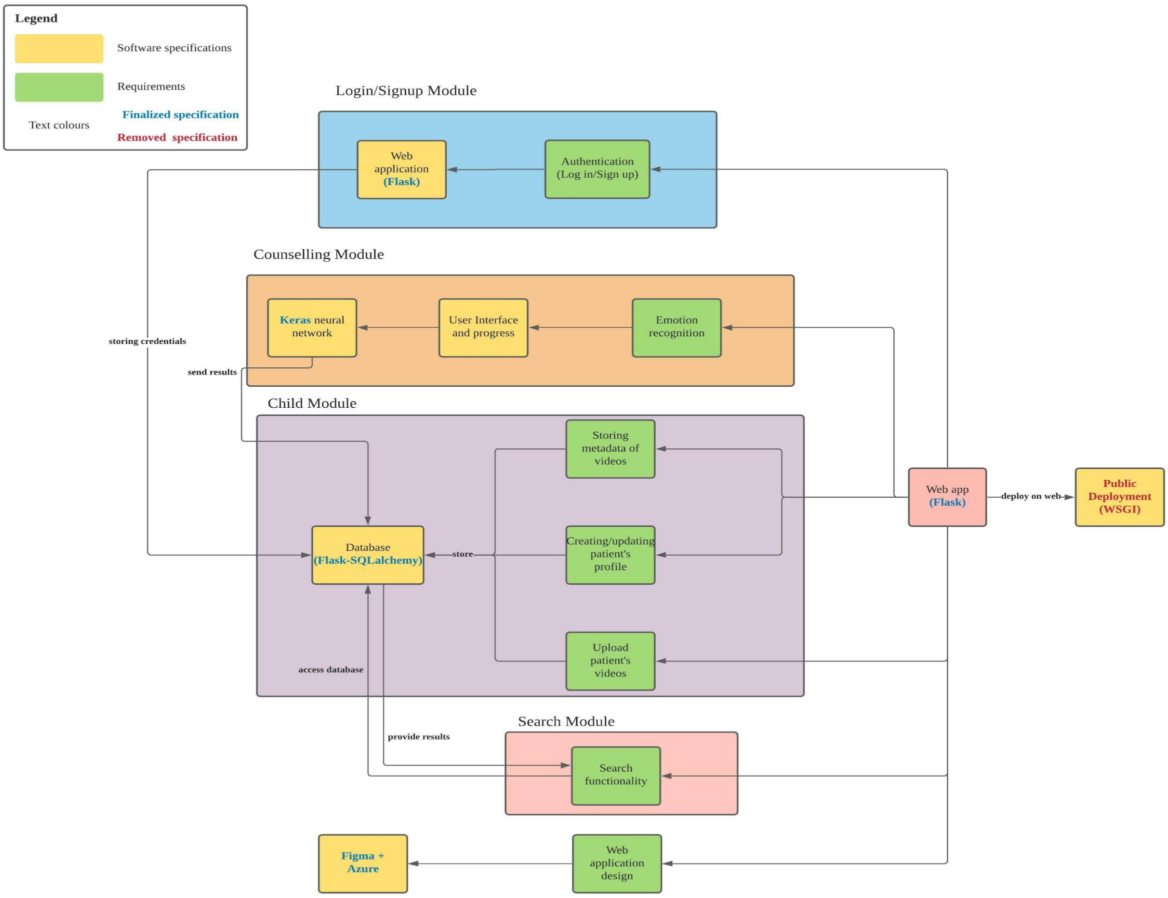

Figure 1. showing the block diagram of web application design.

Notably, each of the software specifications for the requirements of the project have been finalized. In semester one, the specific software for implementing the requirements were tentative and were decided to the best of the team's knowledge at that time. Some of the tentative software tools were not the most appropriate of our use cases and they were not suited for implementing the specific requirements. After trying out new software and tools, we have finally decided to use the tools as described below.

There are four main modules to the project. These include the login/sign up module, counselling module, child module and search module. Each module contains one or more project requirements. The login/sign up module handles the authentication of the user. This module uses the database to save any new user information (such as the email and password) during the registration and to query the database during the login process. This is handled using Flask Forms to receive user input and the Flask-SQLAlchemy to save it into the database.

The counselling module calls the DL script to process the user uploaded video in order to get emotion tags. There is also a progress bar to indicate the progress of the processing. The original plan of using a Tensorflow DL implementation was finalised to a Keras DL network. These were made inline with the changes to the planned architecture as detailed in the literature review. The results of the DL which are the predicted emotion category for each timestamp is stored in the database for later retrieval to display in the web application.

The child module includes requirements such as storing the metadata of the videos, updating patient's profiles and uploading patient's videos. This generated data from the usage of the web application is stored in the database. The main idea behind this module is to reduce the rudimentary tasks of managing patient's data. The database implementation used is the Flask-SQLAlchemy library (which uses SQLite) instead of the planned mySQL resource specification. This is because the Flask-SQLAlchemy works really well with the Flask library since it is an extension of Flask. It is also chosen because it is familiar with the team since it is a Relational Database. It has models (similar to tables) which have one or more columns (attributes) and rows (records) with foreign keys as relations. This familiarity allows all the team members to understand and contribute to the development of this module.

In the search module, users can perform multi-criteria search on videos and emotion tags. Many of the searchable data are generated from the child module. This module will query the database to receive the data processed by the web application and generated by the user. However, as will be discussed in the Outcomes section, the search functionality in the final deliverable is limited.

The web application design, which is the HTML structure of the web application, is done using Figma. Figmna will create basic HTML files of different designs that we created. Combined with Axure, we can export the automatically generated HTML files based on the designs from Figma. This enables people without a solid knowledge on designing web applications to easily convert their prototypes into code which can directly be used in their web applications.

All of these modules are tied together by the Flask framework which manages the routing, displaying of individual pages, creating forms, storing information into the database, querying results to display them, and calling the DL script. Originally, the public deployment feature was an optional feature suggested by our supervisor. This will enable easy access to test the final deliverable. However, due to a shortage of time resources, we are unable to deploy the web application publicly. Instead, the team focused on ensuring that all the required features are present in the web application.

3b) Deep Learning FER Design and software specification

As have been discussed in the literature review, the design has changed from the light residual network architecture from Qayyum & Razzak to a frame-to-sequence approach from Kuo et al. This is chosen to increase the accuracy of the DL network at producing predictions for sequential data such as videos. Figure 2 below shows the frame-to-sequence approach proposed by Kue et al.

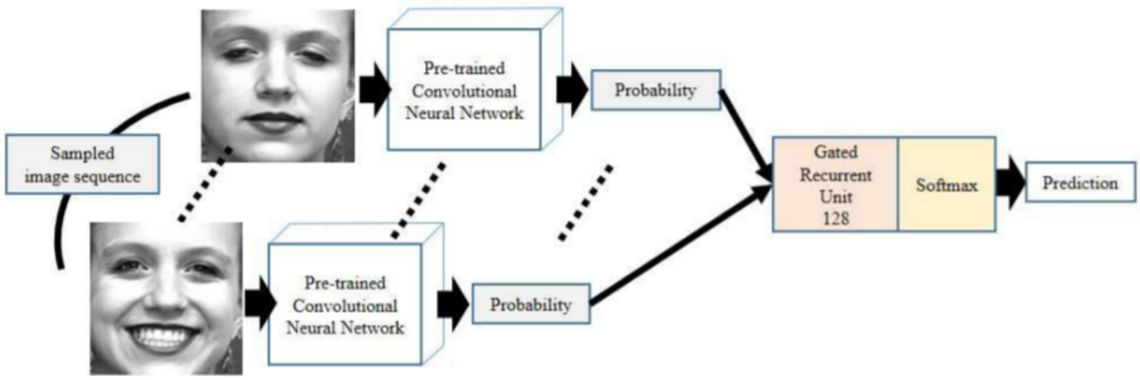

Figure 3. Framework of the proposed frame-to-sequence approach. The frame-to-sequence model takes features extracted by the pre-trained CNN model and uses their softmax outputs for classification.

Figure 2. Frame-to-sequence approach from Kuo et al.

The main idea is to input a sequence of images (i.e. multiple images) rather than a single image. A trained CNN will give an emotion prediction (in terms of probabilities) for each of these images, and then these probabilities will be concatenated together before entering the GRU layer. Then, this GRU layer will learn the temporal aspects of the image sequence, to produce a final output of the probabilities of each emotion category. Thus, there will be a single prediction for a video segment rather than the conventional single image single output obtained in a CNN. The final output corresponds to the predicted emotion contained within the sequence of images.

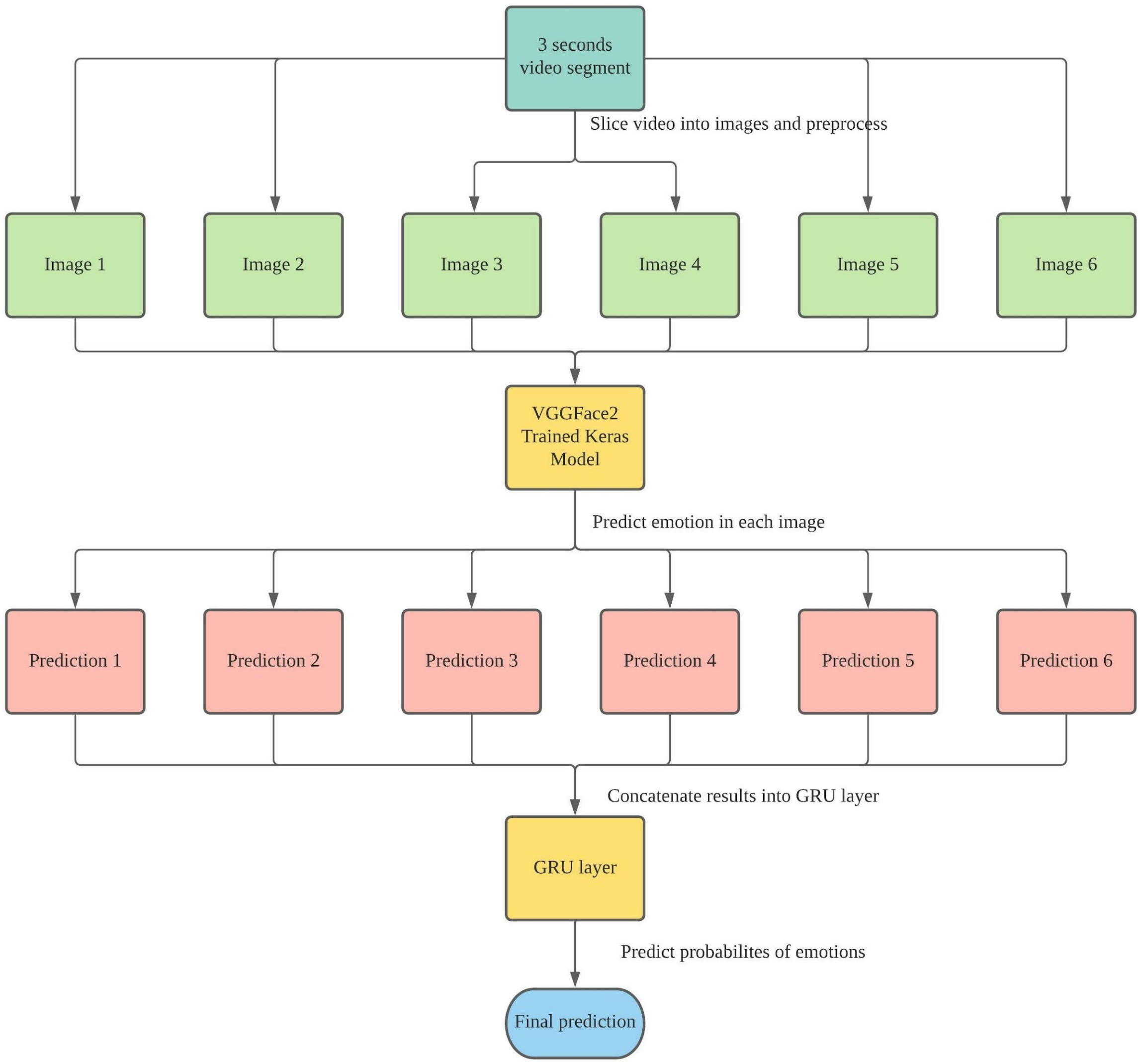

Figure 3. DL architecture for FER inspired by frame-to-sequence approach from Kuo et al.

The input to the DL is a video segment. This video segment can be of any arbitrary duration, however a good choice is to select an appropriate granularity for prediction. A fixed duration of three seconds for a video segment means that there is a single prediction for that time span. We have decided that three seconds is a fitting time duration that is neither too long nor too short. This is dependent on the specific use case intended by the stakeholder.

The video segment is then sliced into six images. Preprocessing steps are also done to ensure that the input images are uniform and predictions are accurate. Examples of preprocessing steps besides image slicing are face extraction and image resizing. The aim of face extraction is to remove the background noise from the image so that the DL only learns the expressions from the face which leads to better performance. Resizing is needed to feed the image into the trained model since it requires a specific image dimension. Thus the DL can purely focus on the patient's face and not any other extraneous variables.

We used the trained VGGFace2 from Cao et al as our trained model to produce predicted probabilities and feed it into the GRU layer. A single outcome of probabilities will be produced. To get the highest confident prediction of the emotion category, we can take the maximum of the values. This implies that the DL has recognized an emotion that occurred within the chosen time frame.

Another interesting feature is that the DL only gives a prediction if it is somewhat confident about the predictions. This is achieved by putting a threshold value on the output. Currently, the script implementation has a threshold value of 0.3, which means only if the highest probability of an emotion category is above 0.3, will the script produce a result based on that category. This was intended to be customisable by the user in the neural network interface, in the case where users might have different preferences on the confidence level of the output. But due to the limitations of time, this was not implemented. (See Requirement ID 009 in Requirements Traceability Matrix)

3c) Activities carried out

Here are the main activities carried out in order to implement the design above throughout the semester. These are the list of activities done to create the web application:

1. Researched on the most suitable website development tools and decided to go with Flask due to its accessibility and wide-range of functionalities that enables users to deploy.
2. Created multiple routes to different web pages such as Login, Register, Dashboard, etc.
3. Constructed a database using sqlalchemy and SQLite to store information that will be necessary for the website i.e. video details, username and password, patient id, etc.
4. Implemented the various basic functionalities of the website using Flask and HTML, such as the user authentication, forms to be used, search and pages that displayed information from the database.
5. Built a video uploader webpage where the user is able to upload the videos that are to be processed.
6. Connected the python script containing the Deep Learning module with the Website application and configured the design (HTML/CSS) based on the output.
7. Finally, we exported our design UI/UX on Anima into HTML & CSS code which we used as a reference and incorporated this design onto all our pages to maintain a consistent look.
8. Conducted a Q&A test on the entire website based on the user's experience. This information was retrieved and modified to improve the overall flow of the website.

For the Deep Learning (DL), the activities carried out are:

1. Researched DL architecture and finalised the design based on Kuo et al.
2. Data sourcing by obtaining the LIRIS-CSE from Khan et al.
3. Data preparation including video slicing, face extraction, resizing, duplication and augmentation.
4. Set up the neural network in Keras by building the pipeline. The pipeline begins from feeding preprocessed images into the Functional API network in Keras.

5. Trained the neural network in GPULab which provides cloud GPU resources for faster training time. Saved the different models with different accuracies and metrics.
6. Tested the performance of the saved models.
7. Finalised it by putting all of these into a script which is used inside the web application in the Counselling Module. Ensure that the web application can call the DL script and results are stored in the database for later retrieval.

4) Outcomes

In this section, an overview of the outcome of the final deliverable will be described briefly. Next, the results and limitations of the implementation will be discussed with reference to the original requirements as stated in the Requirements Traceability Matrix (RTM) that is attached in the appendix. Then, a critical evaluation is performed on the limitations of the implementation and then the justification of any important decisions made in relation to the requirements of the project. Finally, there will be a conclusion of all the results obtained and possible opportunities for future work.

4a) Overview of final deliverable

The web application that has been created is a locally hosted web application that allows users to access different functionalities such as authentication, updating a patient's profile, uploading videos, processing videos and multi-criteria search. All of these different functionalities are accessible from a navigable dashboard which is user friendly. This web application is coded in Flask which is a Python micro-framework for web development with the help of a few flask extensions to enable management of the database (Flask-SQLAlchemy), form validation, user authentication and uploading files. These are implemented according to the Methodology section of the report.

For the Deep Learning component, we have implemented a fully functional Keras neural network which can be called by the web application. This fits into the counselling module (described in the Methodology section) which is part of the project requirements. Users can process their video to produce tags with the click of a button and any results from the DL is stored into the database.

4b) Results, limitations and project requirements.

The software development process aims to meet the requirements as stated in the Requirements Traceability Matrix and User Acceptance Criteria.

1. **Requirement ID 001**: Authentication to access website application.

   The authentication of users is met. Users have to register by creating an account before using the web application. They can also login using their details. This will ensure the security of the data is not compromised as the patient's data are also displayed in the web application. This requirement is met.

2. **Requirement ID 002**: Authentication for forgotten password.

   While this was implemented in code, the functionality does not work due to not owning an SMTP email account. An SMTP email account is required for flask to send emails to users, and this functionality would be needed to send the token to redirect the user to reset their password, however, such an account would need to be purchased. Due to poor planning here, we did not have the time to buy and claim the funds for the email account. Hence this requirement is not met.

3. **Requirement ID 003**: Website application dashboard.

   Web application dashboard has been created. Users can access different functionalities such as looking up patient records, add new patient records, managing their account, uploading video and other functionalities. The dashboard is the main page where users can navigate to the different individual pages to access the functionalities required. This requirement is met.

4. **Requirement ID 004**: Updating/modifying patient's profile.

   In the Add Records page, users can add new patient records. Data such as name, age, gender, and guardian information can be uploaded by filling up the form in that page. These values are then stored in the database and can be retrieved later in the Records page. On each record page, the information can be updated at any time. Users can also delete a particular record if needed.

Additionally, on each record page, any uploaded videos linked by patient ID will be listed and linked to as well. This requirement is fully met.

5. **Requirement ID 005**: Storing information such as profile for patients in the video.

   Storing information generated by the web application is implemented using the Flask-SQLAlchemy in the Child module (see the Methodology section). Any data generated from the different functionalities are stored in the database. These stored data are used to be displayed in the many HTML files as needed. This requirement is met.

6. **Requirement ID 006**: Prompt to upload video files from Desktop/Third-party.

   Users can also upload recorded counselling sessions which are saved in their desktop into the web application in the Uploads page. All of the recorded videos can be reviewed in the Video Database page by the user. This requirement is met.

7. **Requirement ID 007**: Store metadata of the uploaded video such as date/time, name, status.

   Metadata associated with the uploaded video includes the date time of uploaded time, name of the video and the status of whether the video has been tagged by the DL backend. All of this metadata is displayed in the Video Database page. This requirement is met.

8. **Requirement ID 008**: Neural networks which detect and classify emotions. Used as the backend program.

   The neural network has also been implemented using Keras. In the Video Database page, if the video has not been processed (as indicated by a button in the records), users can choose to call the DL neural network to process it to automatically produce tags of timestamps where emotions are detected. Results are displayed in another page and users can skip to specific timestamps by clicking on the displayed timestamps. This neural network also saves the output in the database to display it appropriately in the output page. This requirement is met.

9. **Requirement ID 009**: Neural network User Interface.

    For the neural network interface, a simple progress bar has been implemented to indicate the progress of DL in categorizing the specific video that was chosen. Despite that, this progress bar comes in a pop-up window, rather than being embedded in the web application. This means that the progress bar is not fully functional and may be sometimes slow and not responding. Thus, it is not a perfect and satisfactory implementation. This requirement is partially met.

10. **Requirement ID 010**: Multi Criteria Search Functionality.

    Due to time constraints we could not implement the multi-criteria search functionality. Search functionality was implemented only as a simple string search as multi-criteria did not work and there was a lack of time in trying to debug it. However the string search can search across the different aspects of the records such as first name, last name and country. Thus while not fully satisfactory, a rudimentary version has still been implemented. Hence this requirement is partially met.

4c) Critical Discussion

A. On UI/UX design

One of the focal points of the project is to create a web application which is user friendly and not just containing all the required features. Thus, the UI/UX designs are important to ensure usability and accessibility for all users. In semester one, the team has planned to use Figma to create prototypes of the web application design. This allows the team to focus on ensuring good UI and UX designs. The team has tried to follow these example principles such as ensuring users have ease of control of the interface, comfortability of interaction, and making user interfaces consistent. (Babich, 2019)

By using Figma, this allows the team to create prototypes of the web application design. Combined with the use of Anima, this allows the team to export the created prototypes into HTML code that can be readily used as the design of the web application. However, one drawback of this method is that the exported code does not contain any of the simple functionality we have expected. Interactions such as visual feedback and cues like highlighted elements are absent from the exported HTML code. The exported code purely contributed as the structural element of the web application and was missing anticipated functionality. Thus, all of the user interactions had to be manually coded in the flask application.

Another issue with the exported HTML code is the complexity of the generated code. Not only does the generated code contain the HTML structure, usually each exported prototype consists of a block of javascript code that was not functioning. This block of code made up more than half of the generated code. These blocks of code were meant to provide interaction with the HTML elements on the web application, but with vaguely named functions and no documentation, there were no means available to the team to comprehend and reuse it in a meaningful way. Therefore the team has found working with the exported designs difficult and the goal of developing a user friendly interface based on the created prototypes in semester one was not met.

Another implication of the above issue coupled with our lack of knowledge meant that many of the HTML pages have to be coded from scratch. As expected, the appearance and functionality of the web application is simply not comparable to the prototypes, since we did not have the time to bring our skills up to par to replicate the Figma design. An example of this is the incomplete Requirement ID 009. Although there is a pop up window which displays the progress of the DL, it did not meet the expectations of being a component of the web application itself. This was due to the incompatibility of the Python DL script and the Javascript which is used for web applications. With limited resources, the team has not met this requirement.

B. Poor accuracy of DL

As shown previously, the accuracy of the DL is unsatisfactory. This is in spite of the fact that we have employed the use of a pre-trained network to do fine tuning. A post mortem analysis during the sprint review in Week 8 revealed a few potential issues to why the DL is underperforming.

One major issue is the overestimate of the amount of time that we have to implement the project. We were too ambitious to develop our DL network. To create a more accurate model, we have decided to adopt a more novel approach following Kuo et al using the frame-to-sequence approach. This however, complicates the implementation process as we have to learn many new things that are relatively easy to implement.

For example, our DL architecture which requires six images as input makes the preprocessing stages much more complex. Not only do we have to do the slicing, resizing, face extraction, we also need to do data duplication in the scenario where a video does not have a multiple of six slices. The splitting of the data into test, train and validation sets also have to be done manually using some mathematical calculations to form the correct batches. To feed the input data into the network, custom made generators are also written to generate batches for the fit function.

Besides that, the unconventional DL architecture also requires us to learn the Functional Keras model as opposed to the conventional Sequential model which we are exposed to in our Deep Learning unit. All in all, these steep learning curves made our DL implementation take longer than expected as we had to learn all of these on the go. Although we have given our best efforts, there exist many bugs in our code which made the training stop abruptly or fail to train properly. These made it difficult to properly test the network as well. This is one of the factors leading to the poor performance of the DL network.

Another factor is the size of the pre-trained network. The pretrained network that we have used for transfer learning is trained on a very large number of images (around 3.31 million images) and gives accurate predictions. Unfortunately, due to the size of this network which is around 29 million parameters, our original plan of using the free GPU provided by Google Colaboratory did not work simply because it took too long to train. As a comparison, we used LIRIS-CSE with only 26 thousand images which is significantly smaller than the original amount of 3.31 million images used by the researchers. This could imply that the amount of data we have is not enough to properly train the huge network. In the risk register (see appendix at risk number 1), to account for this risk, we plan to source for more data. But after some deliberation we think that it is not the amount of the data that is the issue but the size of the pre-trained network that we used is overkill for our relatively small project. We have tried to experiment with a smaller pre-trained network, but due to lack of time, we had to finalize the implementation of the DL to focus on the web application.

The factors above are not something that could be fixed by any proper project management strategies as it is a lack of understanding of DL and being too ambitious with our planning. We have reflected upon this and have learnt that although it is good to try out new things, we have to give ourselves leeway in the form of time to account for difficulties and delays that we may encounter due to implementation difficulties.

4d) Justification of decisions made

Due to the discussion on the UI/UX design above, the team has decided to code the HTML for the web app from the bottom up. This meant that the final outcome of the web application is significantly different from the original design in semester one. Despite that, the team has decided that trying to meet our requirements was more important than strictly following our planned design. As it stands, if the exported designs were used, many of the buttons, navigation and visual cues were not working and thus would be jarring to the user.

From the discussion on the low accuracy of the DL, the team have settled for a lower accuracy network. After some discussion close to the mid semester break, the team has decided to put a halt on the development of the DL and focus on integrating the DL script with the rest of the web application. Although the accuracy of the DL is important and could be improved further, it is even more important to incorporate it with the rest of the web application to fulfill the requirements.

All of the decisions that have been made were made to ensure that the final software deliverable satisfies as many requirements as possible with the best quality.

Due to the reasons as mentioned above, as well as time constraints, the team had to compromise on certain aspects of the project such as the look and feel of the web application and also the performance of DL network. The team has decided that a more coherent product at the end would be much more important than completing only a few project requirements which are not integrated and disjointed.

4e) Discussion of all results and future work

As a summary, many of the basic project requirements have been satisfied. But there exist many places to improve on such as the UI/UX of the web application and the accuracy of the DL. From our reflection, one major amendment that can be done is to have a better estimate on the project resources required to implement the requirements. For example, a better understanding of the software tools and its limitations can ease development. Time resources can be better handled with more scrutiny on the workload requirements of each task and activity. The change in the DL scope from semester one can be analysed based on time of work required. Therefore, the triple constraint of project management is a very critical part of any project.

5) Software Deliverables

5a) Summary of software deliverable

The software deliverables includes the Python source code which uses Flask and all of its related libraries to create the web application. The specifics of the requirements covered and the software specifications are covered in the Methodology section. The bundled source code also includes the Deep Learning model which is stored as a .h5 file. By running the web application through the console (as described in the user guide), users can access the locally hosted web application which does not have any other dependencies to be installed except for the mentioned python packages and modules. This standalone application provides the functionalities as described in the Methodology section.

Below are some screenshots of the web application and a very general overview of the key functional requirements that have been implemented. Note that this is not meant to be a comprehensive showcase of all the completed functionalities. A more thorough explanation will be provided in the user guide document.

Figure 4. Web application dashboard

In Figure 4 above, it shows the dashboard when users managed to login successfully to the web application. There is some simple navigation in the application to take users to different pages with their own functionalities as described in the outcomes section.

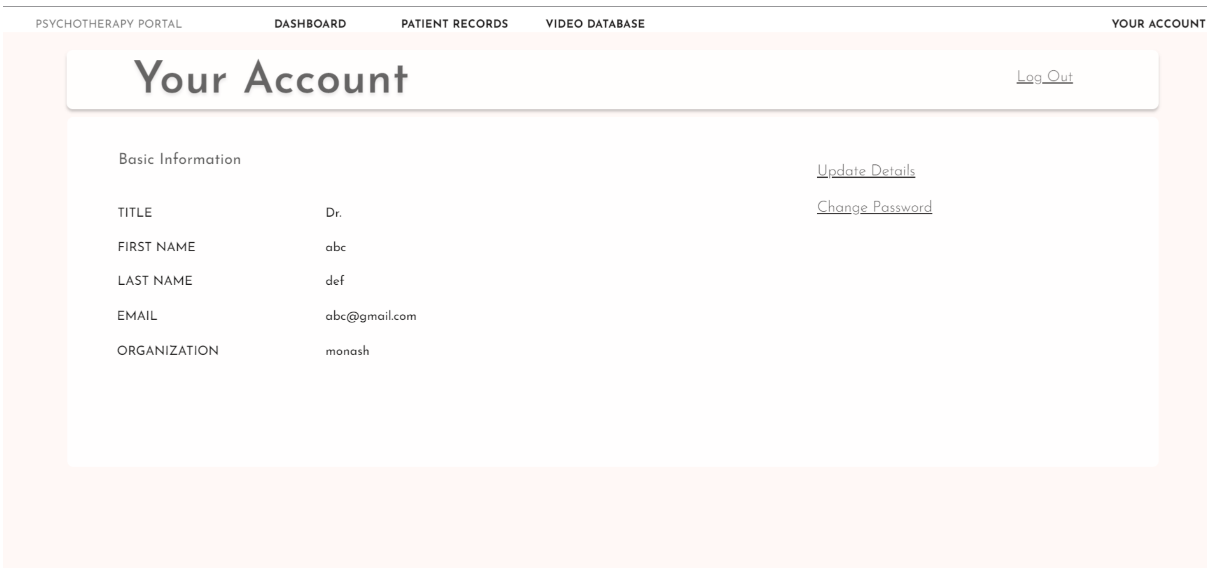

Figure 5. Profile details

Figure 5 above shows the created account that has been used to log into the web application. Users can change and manage their details here.

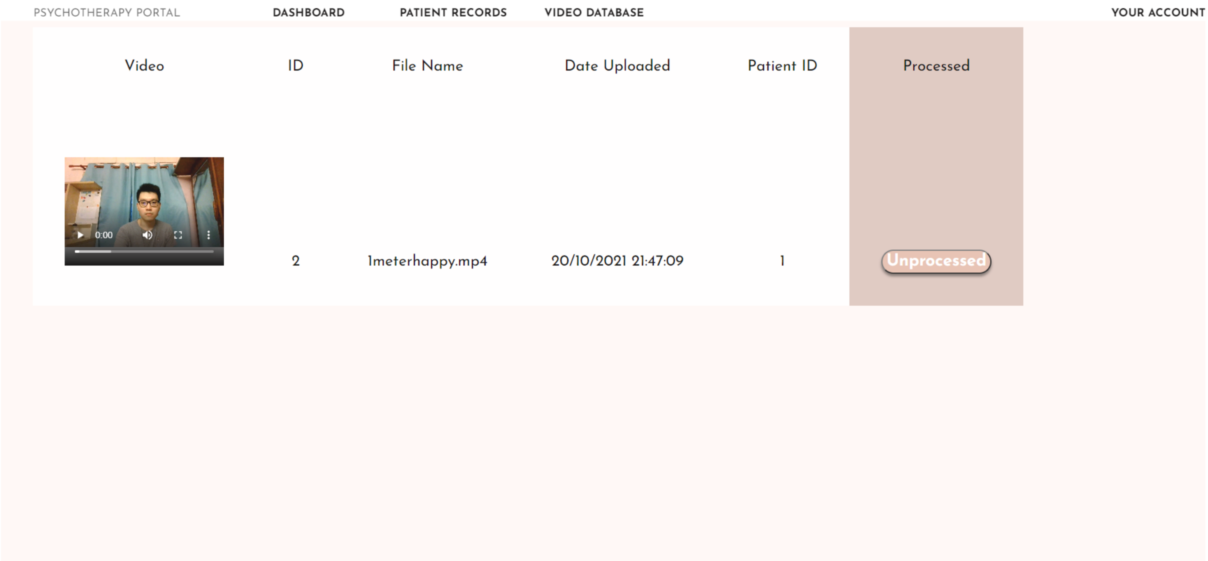

Figure 6. Video Database containing unprocessed video

In Figure 6, once the user (i.e. psychotherapist) has created a patient, they can upload recorded counselling sessions of that particular patient into the web application. Some simple metadata are stored such as the file name, date of upload and the patient id associated with that video. If the video is not processed as shown in the rightmost column, the user can choose to process it to call the DL network to find the detected emotions in the video.

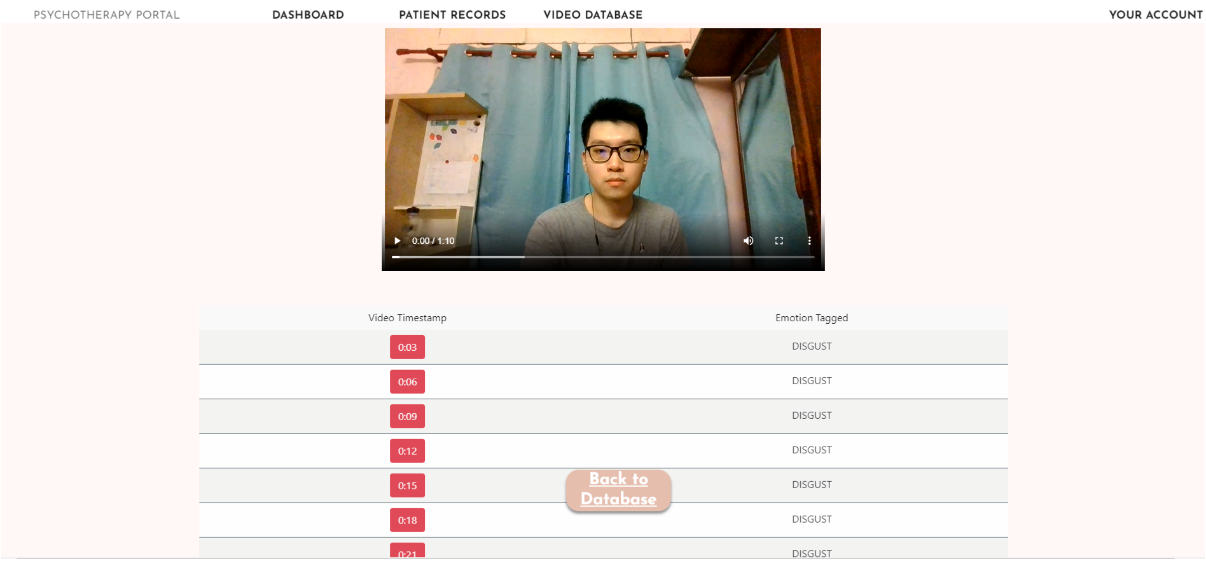

Figure 7. Processed video result

Figure 7 above shows the results after processing the video. The buttons for the video timestamps will allow users to jump to the particular time where an emotion has been detected by the DL FER network. It also shows the emotion tag as predicted with that specific time duration. This will enable users to revisit important durations of the recording without having to manually search for them.

5b) Summary and discussion of software qualities

Software qualities are handled through software quality testing after each part of the software has been completed such as user authentication and deep learning network integration. Based on the testing, our project mostly fulfills the robustness, scalability, usability and security requirements.

A. Robustness/Performance

In terms of performance, our deep learning model is considered low in accuracy as it only has 50% accuracy. The deep learning model is also biased towards certain categories and does not capture nuanced expressions, only exaggerated ones. However, our web application is easy and fast to navigate between one page to another. Though users are forced to wait for the processing of the video using the DL script and cannot do any other tasks.

B. Security

Only authorised users can access the application where they have full access to the videos uploaded, patients' information and other information that are added by corresponding users. However, the information is not encrypted in the database, which could be an issue if someone unauthorized tried to access it. Since the project is not publicly hosted, this is not a very big issue, however more secure steps have to be taken before the data is stored in the database in the case where the web application is deployed.

C. Usability

The application has a clear objective of what it is used for since the title of the application is stated on the main page. Furthermore, it is easy to use that no prior knowledge is needed considering its intended functions and the user's need. One limitation is that the UI/UX is not as polished as originally planned in semester one as discussed in the outcomes section.

D. Scalability

Since the application relies on local storage, its performance depends on the user's CPU performance. The smaller the CPU memory is, the longer it will take for the application (Facial Emotion Recognition part) to process the video. In Addition, since the application is locally hosted, all of the files required to run the application needs to be stored in the client side. This could take up space and is highly dependent on the user’s memory storage in their machine. This could have been avoided by publicly hosting our web application where our machine is the server. Thus, the user does not have to install anything locally. However, due to time constraints this is not implemented and is a place for future work.

E. Documentation and Maintainability

To ease reading and comprehend the code, there are documentations written within the code. This documentation is intended for both developers and non-developers to understand the what and how of the code as well as to ease when it comes to fixing the bug.

To document the software testing, there is a template for quality assurance testers to fill in and the outcome is optional to be or not to be discussed with the developer for further improvements. If further discussion is needed and some fixes are done, another testing will be done to ensure maintainability of the software.

5c) Summary of software deliverable

In summary, the final deliverable is a web application that employs a deep learning script for Facial Expression Recognition (FER). Since the web application is not publicly hosted for now, to run the web application, users have to enter a few commands in the command prompt or console to execute the program (More details will be provided in the user guide). The locally hosted web application will run the Flask program in the Python file. Then it will handle the routing to different pages using app.route, displaying the individual pages using the HTML files using render_template function, using Flask Forms to accept user data, and using Flask-SQLAlchemy to store user inputs into the relevant database. Users are able to invoke the DL script which is packaged together with the web application files to process an uploaded video to detect facial expressions.

5d) Sample Source Code

In the appendix, a sample code is given. In this section, we will describe briefly about the source code. The code given is in multiple blocks since it is not consecutive lines of code. These blocks of code intend to show how the requirement ID 008 (Neural network detects and classifies emotions) are fulfilled.

Code block(1) shows the processing which occurs when the users want to access the results of the processed video. If it's not processed yet, the web application will find the video in the local directory and call the main function which processes the video using the DL network. Once the DL script produces results, it will store it inside the database and commit.

Before displaying the HTML file, it will query the database for the emotions tags (results from the DL) and render the results along with the HTML page.

Code block (2) shows how the results are displayed in the HTML file. It has a for loop which iterates through the output (list of emotion tags), then displays the timestamp as an interactive button which performs a jump action and also displays the result emotion tag (such as happy, sad, disgust, etc).

Code block (3) shows how the jump action is performed. It gets the html element for the video and the button which is displayed. For all the buttons corresponding to the timestamps (the results may have a button for the third second and sixth second of the video), it will create an event listener to wait for the user to click on it. Upon clicking, the video will jump to that time in the video based on some calculations.

6) Critical discussion on project execution

In this section, we will analyse the project execution compared to the initial proposal in semester one. The projects' execution will be analysed from the perspective of the project methodology, managing project resources, accomplishing the targeted requirements and a conclusion on all of the deviations from the initial plan during the project's execution.

6a) Project Methodology

In terms of project methodology, the team has followed the software development methodology as described in semester one very closely. Sprint planning and sprint review are conducted on a regular basis, with meeting minutes being recorded and if there is a need; video recording of the meetings are done for absentees. Sprint retrospective with our supervisor is also carried out once every two weeks to get feedback from the stakeholders about the progress of the project by showing a proof of concept. Doubts are clarified, and suggestions are given during this meeting.

As a whole, the team has stuck to the original Small Scale Scrum methodology which focused on incremental approach based on feedback from the stakeholders. We have found that adhering to this software development methodology was useful to keep everyone on track. There was constant progress each week which meant that the amount of backlog tasks were limited. Although there were some hiccups due to postponement of meetings and the pressure from constant goals setting in sprint planning, the team has found this approach desirable compared to an unstructured software development method or even the waterfall approach.

6b) Resource Requirements

On the topic of resource requirements and software specifications, there were some significant alterations that were made in terms of software tools used for development. Examples include changing the DL framework from Tensorflow to Keras, usage of exported code from Figma and Anima, using SQLite under Flask-SQLAlchemy, video data storage, and machine learning hardware. The first two examples were covered under the Outcomes section. Other changes were made due to unsuitability of those tools after the team had tried using them. Most of this stems from the fact that the team did not have much knowledge and expertise in the aforementioned software tools.

As mentioned in the Discussion of all results and future work section, a better understanding of these tools can prove useful before working with them. The team could have tried to research deeper into said tools before committing to using them in semester one. This could have saved time spent on researching new software tools in semester two. With that being said, the current tools used to deliver the final product are fitting for developing the project requirements and use cases. A major reason why the changes in the software tools worked is due to the flexibility and adaptability of the team mindset on learning new software tools. The team has realized the importance of being able to adapt to new tools as quickly as possible when decisions are made to use them.

6c) Cost management

From the perspective of project cost management, in semester one, the team did not plan to use any tools or services that cost money. This is reflected in the non-existent paid tools and services in the Project Proposal in semester one. However in semester two, after some deliberation the team has decided to procure some code exporting software such as Anima and a cloud GPU service called GPULab for the development of the web application and DL training respectively. Originally, these services were not considered due to the cost and the team was relying on the initial tools and expected them to be sufficient for the development. Although these tools were usable, they were unexpectedly unsuitable for the use case.

For example, the training of the DL network was supposed to be done in Google Colaboratory which provided free GPU service. Unknown to us at that time, the GPU service is limited based on the user's utilisation and will terminate after frequent use. Not only that, it is also not as fast as predicted which forced the team to look for other paid alternatives. This could have been avoided from the start of the project by conducting a more thorough project cost and procurement management. Nevertheless, these tools worked in terms of realizing the project by allowing the team to export HTML designs and train the DL network at a much faster pace.

6d) Achieving project requirements

In terms of meeting user's acceptance criteria and achieving the project requirements, the final deliverable can be said to be satisfactory by meeting many of the set requirements as described in the Outcomes section. Despite that, the requirements are fulfilled only at the basic level, and further improvements could be made to enhance each of the functional and non-functional requirements. Examples are a smoother and more customizable neural network interface, a more informative web application dashboard and a more detailed multi-criteria search.

Since the scope and requirements of the project were not amended from semester one, the team has found the workload to be too optimistic which led to imperfections in the final outcome. With a better understanding of the scope and resource management, the team could achieve an even better final product. This could be done by planning the scope based on available resources such as time and human resources.The team could also communicate these issues with the stakeholder to possibly narrow down the scope to only the vital functionalities of the project with the purpose to improve quality.

6e) Conclusion

In conclusion, there were some parts of the project that abided by the initial proposal developed in semester one. This was beneficial to the team because the planning that was made throughout the previous semester was methodical and systematic. Many of the decisions were well thought out and it paid off to stick to the established decisions. On the other hand, some of the decisions had to be altered because as we progressed through the project, the team gained more knowledge and realized some of the initial plan does not fit the implementation very well. Changes are required to properly implement the functional requirements planned.

7) Conclusion

In conclusion, the project was conducted with some successes and failures that we have recognized. The team has followed the initial proposal as suggested in semester one to the fullest possible extent, and have made a few changes to it in appropriate places. There were a few changes made to the literature review which gave us a deeper understanding of the background knowledge, especially the DL component of the project. Besides that, the requirements and scope of the project were untouched, and as a result of that, the methodology has not significantly changed from last semester. Some changes were made to the software specifications to better suit implementation choices. Detailed analysis of the project from multiple perspectives such as the final outcome to the project execution were performed. This report encapsulates all of this information and includes the team's introspection on the relative successes and shortcomings.

As a result, a web application has been successfully developed which aimed at reducing the manual workload of psychotherapists. The web application was developed with a heavy emphasis on accessibility and usability of users. While there were some unmet requirements, many of the functionalities and features that were planned were completed, especially in regards to the core functionalities of emotional tagging and patient record keeping which by themselves would already increase the efficiency of the psychotherapy work. However, there is certainly a need to improve on these functionalities and features in future work. The team finds the final product as a promising starting point for any future development into a useful application.

Since our project was catering to child psychotherapists, the DL was also created and trained on databases containing children videos such as LIRIS-CSE. However due to the lack of training time and complexity of the DL architecture, the accuracy of the DL network was not as optimal as hoped. Even though this might be a limitation, the DL script can be easily swapped out with a better performing model if required. This is because the input and output is fully synchronized with the database in the web application.

As a summary, we have found the project to be a success. The team hopes that this simple web application may prove to be useful in the future with added improvements and refinements. We do believe that there is a need for online psychotherapy services especially with the increased demand for such services in recent years, and any application that can streamline the process of managing the data used by mental health practitioners is extremely beneficial. As mental health is important to our well-being and is now gaining more traction, this project will bring a positive contribution to society.

8) Appendix

Sample Source Code for Software Deliverable section

A. Code Block (1)

```
@app.route("/processedVideo/<int:videoID>")
def processedVideo(videoID):
    #Display the Processed video page:
    vid = VideoFiles.query.get_or_404(videoID)
    if vid.videoEmotion == None: #If did not process the video before
        #Perform the DL output here. Shld print here then render template with
the output
        try:
            videoDirectory =  os.path.join(os.getcwd(), 'FYP', 'static',
'uploads', vid.videoName)
            print(videoDirectory)
            predictedresult = main(videoDirectory)

            #Store the results in the database
            vid.videoEmotion = predictedresult
            db.session.commit()

        except Exception as e: #print error message
            db.session.rollback()
            raise
            print(e)

    #Query the results from the database
    queriedResults = VideoFiles.query.get_or_404(videoID).videoEmotion
    print(queriedResults)

    # send the list of results to display the results for each vid:
    return render_template("outputDL.html", video = vid, ouput =
queriedResults)
```

B. Code Block (2)

```
<tbody>
    {% for vid in ouput %}
    <tr>
        <td class="text-center">
            <button id = "jump{{ loop.index }}" type="button" class="btn
btn-danger" value="{{vid[1]}}">{{vid[1]}}</button>
        </td>
```

```
        <td class="text-center" style="text-transform:
uppercase;">{{vid[0]}}</td>
    </tr>
    {%endfor%}
</tbody>
```

C. Code Block (3)

```
<script>
    // get element based on the id tag myvideo
    var myvideo = document.getElementById('myvideo'),

    // jumplink = document.getElementById('jump');
    jumplink = document.getElementsByClassName("btn btn-danger"); //Get a
collection of html elements by class name

    // for each of the clicks that take place, create an event where the video
is jumped to the timestamp
    for (var i = 0; i< jumplink.length; i ++ ){

        jumplink[i].addEventListener("click", function (event) {
        event.preventDefault();
        myvideo.play();
        myvideo.pause();
        myvideo.currentTime = hmsToSecondsOnly(this.value);
        myvideo.play();
        }, false);
    }

    // convert the time to only seconds:
    function hmsToSecondsOnly(str) {
        var p = str.split(':'),
            s = 0, m = 1;

        while (p.length > 0) {
            s += m * parseInt(p.pop(), 10);
            m *= 60;
        }

        return s;
    }

</script>
```

Risk Register

Prepared by:		FIT3161_MA_8			**Date:**	**24/5/2021**						
No.	**Rank**	**Risk**	**Description**	**Category**	**Triggers**	**Root Cause**	**Potential Responses**	**Risk Owner**	**Probability**	**Impact**	**Status**	**Score (Impact * Probability)**
1	4	Dataset does not provide adequate training for a reliable Deep Learning (DL) model.	The dataset fails to provide enough input in terms of quality and quantity to train an accurate and reliable Facial Expression Recognition (FER) Model.	Technical	Poor performance when testing.	Dataset is not diverse enough and contains too few samples or low quality.	Look for other sources of data to supplement the main dataset (LIRIS-CSE) if required.	Software Develop er	6	7	Open	42
2	3	FER model's performance does not relate well to real life scenarios.	FER model performs well during internal testing but does not perform well on real life input given by the users.	Technical	Model performs well on validation and testing using our dataset, but performs poorly on unseen user input.	Samples in the dataset may not be similar to the real life samples. May be caused from different backgrounds , lighting conditions, angle, others.	Ensure the dataset contains enough variety to cater to different types of scenarios. Include a user manual to explain video setup for FER to work best.	Quality assuranc e	6	8	Open	48

3	2	Long training time for FER model.	FER model takes longer than expected to train which delays the development schedule.	Time management	Lag in scheduling and fail to meet expected time of completion or milestones.	Low specifications of machines that the DL is trained on or training is more extensive than anticipated.	Use of extra GPU's such as Google Colab or purchasing cloud GPU/TPU for quicker training.	Project manager	7	7	Open	49
4	8	Web application has poor usability.	The web application is too complex for users to use.	Usability	Negative feedback from users regarding the web application navigation through dashboard and pages.	Web application is poorly designed or has not kept usability in mind.	Get constant feedback from the supervisor and perform user acceptance testing.	Quality assurance	4	7	Open	28
5	6	High network traffic	Too many people are accessing the web application which creates a bottleneck.	Network	Responses from users about failure to access the web app due to traffic overload.	Servers are not built to handle that volume of traffic.	Outsource or create backup servers to handle unexpected increase in traffic.	Software Developer	4	8	Open	32
6	7	Platform or device uncompatibility.	The web app cannot be run well on certain devices (such as mobile, tablets) in terms of visual performance or	Technical	The poor format and structure of the web app during testing or feedback from	The web app is not supported on the device.	Perform testing on different devices or limit the scope to only provide computer browser compatibility.	Software Developer	5	6	Open	30

			functionality.		users.							
7	1	Workload is too optimistic	The workload is underestimated which leads to low completion rate or slow development.	Time management	Scheduling milestones are not completed within the estimated time.	Lack of understanding on the scope of the product requirement	Allow flexibility in the schedule. If the team cannot finish the project within the time frame, inform the supervisor to reduce the scope of requirements.	Project manager	7	10	Open	70
8	5	Changes in project scope	The product owner may change the scope requirements of the project.	Project management	Communicated by supervisor	Product owner thinks the scope of the project should be changed.	Discuss the scope of the project and update the project tasks, scheduling and responsibilities.	Project manager	3	7	Open	35

Requirements Traceability Matrix

REQUIREMENTS TRACEABILITY MATRIX					
Project Name:		AI-assisted Online Psychotherapy Portal			
Project Manager Name:		Mohit Hotchandani			
Project Description:		Web application that allows upload of videos, performs analysis to tag emotions, and viewing based on search criterias.			
ID	***Requirements (Functional or Non-Functional)***	***Assumption(s) and/or Customer Need(s)***	***Requirement Type***	***Source***	**Status**
001	Authentication to access website application	User needs to log in before using any functionality of website application	Non-Functional requirement	Dr Fermi	Open
002	Authentication for forgotten password	System needs to allow user to change password by sending a password amendment link via email	Functional requirement	Dr Fermi	Open
003	Website application dashboard	Front page of the website application which allows uploading, processing (using neural network backend), searching based on criterias and viewing the search result	Functional requirement	Dr Fermi	Open
004	Updating/modifying patient's profile	Functionality to add/edit/delete information about patients.	Functional requirement	Dr Fermi	Open
005	Storing information such as profile for patients in the video	The website needs to have a system to store videos, profiles of patients in videos, others.	Non-Functional requirement	Dr Fermi	Open
006	Prompt to upload video files from Desktop/Third-party	User needs to upload their video with comprising of their emotions from their desktop or	Functional requirement	Dr Fermi	Open

		third-party			
007	Store metadata of the uploaded video such as date/time, name, status	Metadata about video have to be stored for further retrieval or tagging.	Non-Functional requirement	Dr Fermi	Open
008	Neural networks which detect and classify emotions. Used as the backend program. Implemented using Tensorflow (tentative)	Classifying based on the timestamp. E.g. from the 5th second to the 10th second shows happy, sad, other expressions.	Functional requirement	Dr Fermi	Open
009	Neural network User Interface	Giving users a setup page for the network(if applicable), a visual feedback of progress, and output pages.	Functional requirement	Dr Fermi	Open
010	Multi Criteria Search Functionality	Users can use a different combination of criterias to search for the patient's video(s).	Functional requirement	Dr Fermi	Open

9) References

Babich, N. (2019, Oct 7). The 4 Golden Rules of UI Design. Adobe. https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/

Barak, A., Hen, L., Boniel-Nissim, M., & Shapira, N. (2008). A Comprehensive Review and a Meta-Analysis of the Effectiveness of Internet-Based Psychotherapeutic Interventions. Journal of Technology in Human Services, 26(2-4), 109-160. https://doi.org/10.1080/15228830802094429

Cao, Q., Shen. L., Xie. W., Parkhi, O. M., & Zisserman, A. (2018). Neural network model: From: https://github.com/WeidiXie/Keras-VGGFace2-ResNet50 (Cao et al)

Dilmegani, C. (2020, July 5). Transfer Learning in 2021: What it is & How it works. AIMultiple. https://research.aimultiple.com/transfer-learning/

Khan, R. A., Crenn, A., Meyer, A., & Bouakaz, S. (2019). A novel database of children's spontaneous facial expressions (LIRIS-CSE). Image and Vision Computing, 83-84, 61-69. https://doi.org/10.1016/j.imavis.2019.02.004.

Kuo, C. M., Lai, S. H., Sarkis, M. (2018) A Compact Deep Learning Model for Robust Facial Expression Recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, 2202-22028. Doi: 10.1109/CVPRW.2018.00286.

Li, S., & Deng, W. (2020). Deep Facial Expression Recognition: A Survey". *IEEE Transactions on Affective Computing*. 10.1109/TAFFC.2020.2981446.

Qayyum A., Razzak I. (2021) Deep Residual Neural Network for Child's Spontaneous Facial Expressions Recognition. In: Torsello A., Rossi L., Pelillo M., Biggio B., Robles-Kelly A. (eds) *Structural, Syntactic, and Statistical Pattern Recognition. S+SSPR 2021. Lecture Notes in Computer Science* (vol 12644). Springer, Cham. https://doi.org/10.1007/978-3-030-73973-7_27