**FINAL REPORT**

1. **Agile software development practices**

This project was developed following the principles of Agile Software Development. These principles are developed to increase the agility of software development for professionals, teams, and organisations. Some principles were utilised in the development of this project to reduce errors, increase production rate, and encourage good design practice.

Continuously integrating new working software is our highest priority. This coincides with one of Agile principles. To achieve a specification set by the project, multiple steps of workable code needs to be completed, tested, and integrated to the system. By continuously integrating these working parts, we can measure our progress and understand where we are regarding the completion of our project. The amount of working software which is directly related to our project specification is our indication of progress. Furthermore, by continuously producing and integrating working functionalities, we ensure that the team is working towards the end goal and waste less time in planning and documentation.

We strive to deliver the project specifications using as simple code and techniques as possible. This includes maximising inbuilt functions to create the desired effects. For outputting the image with boxes drawn to highlight the faces, instead of utilising an image editing node.js module, we utilised the canvas tag on HTML. Instead of creating multiple html files to simulate different parts of the program, we used socket.io to simplify communication between server and client and update one html page in real time. By simplifying the way our program functions, errors can be minimised and easily fixed since code is easier to understand

We obtained requirements about the program through the given user stories. According to the user stories, the system needs to utilise IBM Watson machine (US1). An image needs to be uploaded (US2). There must be feedback about the upload process (US3). The result must include the number of faces (US4), the gender of each face (US5), and the average age of each face (US6). The result must also be delivered in audio format (US7). The program must be reusable (US8). These user stories capture the requirements of the program and we were able to produce individual functionalities which becomes the complete program.

The academic environment has a critical impact on the ability to implement some of the Agile practices. Meeting everyday is impossible due to conflicting schedules, assessments, and other commitments such as projects from other units. In addition, although there are certain ideas on how to be more effective and efficient as a team, those ideas are difficult to implement due to the enormous amount of time required for other units which limits how much effort can be spent on this project.

Due to academic environment, our team was not able to adopt daily meetings to work on the project and sustainable development. Other units’ assessments and commitments equate to a limited amount of effort that we can spend on this project. In addition, due to conflicting schedules, it is impossible to meet daily. In addition, we did welcome changing requirements because we did not have changing requirements. Furthermore, not all communication is done face-to-face as often we can only meet once a week. Hence, most of our communication is done online through our Facebook group chat.

Our project could have been executed better with clearer communication between team members. For some parts of the project, there were multiple team members working on the same feature which reduces productivity. Tasks should have been communicated clearer so that no time would be wasted on duplicating a task. Furthermore, there should have been a clearer timeline on how the project should proceed and which functionalities should be completed in what sequence. However, these changes may prove difficult to implement as our members often have multiple assessments due on the same day or just a few days before the project is due. Hence, there is too much unpredictability in everyone’s availability. This weakness in our team can be mitigated through better communication between team members. We should have a better idea of everyone’s commitments and availabilities to understand how much to allocate to each team member to ensure the deadline can be met.

1. **Working in teams**

For this project, we worked in a team of three people. This presents the need to communicate between team members and proper workload allocation.

The bulk of our team’s communication is done through Facebook. We plan and divide the necessary tasks among ourselves then check back with the group when we need assistance or when the task is finished. Our respective tasks are pushed to Github which are then pulled to our respective computers and combined to create a more complete project. The details of the next tasks are discussed and divided between us, and we repeat this cycle until the project is complete. Documentations for our project is done as a team with input simultaneously from all members.

Communication in our group is primarily done through social media and meeting once a week in our weekly lab to discuss our tasks. We check with other team members on regular intervals to determine the completion of our tasks and discuss the upcoming tasks. During our weekly lab, the project can be worked on face-to-face which encourages feedback immediately from other team members. In addition, we can obtain assistance right away from other team members if needed.

The allocation of tasks is based on availability of each member. We recognise that each of us may have vastly different availability and commitments outside of this unit. If a member is overloaded with assessments from other units, we are happy to pick up a bigger portion of the project. In general, the technical aspects of the project are split rather evenly among the group members, and everyone contributes simultaneously to documentation and report. We are content with this allocation as it accounts for individuals’ availability and commitments.

These considerations and practices are not entirely applicable working in a small company. In professional industry, outside commitments are kept to a minimal and the project becomes highest priority. Therefore, there is less need to allocate tasks based on availability of each member, as it is safe to assume all members can put in the same amount of effort and commitment. In addition, the bulk of communication will move from online to face-to-face as the project will need work every day at the workplace. In a professional environment, projects will need to be planned more meticulously than now due to the number of tasks involved and a strict timeline which needs to be adhered to.

1. **Design**

This program was designed with simplicity in mind. All our functions are derived from inbuilt functionalities from HTML or the simplest modules available. By simplifying our program, we minimise the risk of errors and the ease of debugging errors. The program is also split into relatively small modules to promote readability and ease of debugging.

This application is centred around IBM’s Watson. This means extra functionalities can easily be added if it is supported by Watson. Outside of the capabilities of Watson, adding extra features may be difficult when it comes to facial recognition features. However, adding new features regarding HTML or Javascript should not be difficult in general as there is a vast library of modules available to supply the necessary functionalities. Similarly, any extra functionality regarding text-to-speech can be readily added provided Watson supports it. Otherwise, they might be complex to implement.

During the development of our project, we utilised the Node.js module Winston to log information and errors at various stages of the program. By looking at these system outputs, we can determine the stage and state the program is in. In addition, error logging allows us to see whether an error has occurred and the information regarding that error. This is useful in debugging code as we can see where the program encounters errors and what the error is. One way to make errors easier to deal with and our project more error proof is to section user stories sequentially into functions/modules. That way, each module can be tested and debugged independently which provides more apparent information about the functionality in question.

The visual design of our application is perhaps our weakest point. It is currently very coherent and straightforward; however, it is very plain and not aesthetically pleasing. We are happy that the appearance itself is not complicated for users. All functionalities are presented on one page without the need to navigate. On the other hand, the page can look more polished and professional. To make this page more attractive to children, we suggest the use of vibrant colours and interactive elements. Currently, the page leaves a lot of room for new functionalities to be added. New controls to the interface can be added speedily to accommodate new functionalities.