Neil Browning

B00652112

Project Plan

COM547

Contents

[Introduction 2](#_Toc496992632)

[Project Purpose 2](#_Toc496992633)

[Project Context 2](#_Toc496992634)

[Project Aims and Objectives 4](#_Toc496992635)

[Aims 4](#_Toc496992636)

[Stakeholders 4](#_Toc496992637)

[Objectives 4](#_Toc496992638)

[Project Lifecycle 6](#_Toc496992639)

[Potential Lifecycle Models 6](#_Toc496992640)

[Chosen Lifecycle Model 7](#_Toc496992641)

[Project Deliverables 8](#_Toc496992642)

[Timeline 8](#_Toc496992643)

[Project Deliverables 8](#_Toc496992644)

[Project Planning 9](#_Toc496992645)

[Resource Consideration 9](#_Toc496992646)

[Risk Assessment 9](#_Toc496992647)

[Data Management 11](#_Toc496992648)

[Knowledge and Skills Required 11](#_Toc496992649)

[Involving Stakeholders 11](#_Toc496992650)

[References 13](#_Toc496992651)

[Appendix 1 14](#_Toc496992652)

[Appendix 2 15](#_Toc496992653)

[Appendices Files 16](#_Toc496992654)

## Introduction

### Project Purpose

The purpose of the project is to create a tool for managing student attendance and session feedback for students and staff of Ulster University: the uploading of such data; viewing the information in meaningful formats; and reporting back to users via email. This system will replace an existing manual-effort approach in terms of student attendance, and provide new functionality for students to provide individual session feedback.

### Project Context

Ulster University is one of the largest universities in Northern Ireland, with campuses in Belfast, Jordanstown, Coleraine and Magee.

Student attendance is collected by lecturers per session that they hold with students, which is manually entered into a spreadsheet that remains as a flat file with limited capability for searching, analysing or reporting on the data. Lecturers may handle hundreds of students across different modules, and managing this attendance data as it’s currently stored can be difficult and tedious.

This manual process of managing attendance data is inefficient and in need of a more IT-focused approach. The idea for this project was put forward by Dr Alan Brown, associate head of the school of engineering at the Jordanstown campus of Ulster University, who played a role in the existing system of manual spreadsheet entry and proposed that a technical solution would allow for better use of audience data. This audience data processing would start with student attendance, but the existence of such a tool could allow for addressing other opportunities as well – another facet of audience data that this project will address is feedback from students regarding individual sessions. The only formal approach to student feedback currently comes only at the end of modules, where a module survey can be carried out by students who had undertaken that module. However, feedback at this stage in the academic year is already too late, as the students have already finished their work on the module. Any improvements to the content or methods of teaching because of this feedback only benefits the next students who undertake that module. In other words, the feedback cycle is too long; a faster turnaround is required for students to see more benefit in their studies.

The author chose to undertake this project for a few reasons. On the student attendance front, the author believes it’s important that this data is stored accurately and is less prone to human error by manual processing. More importantly, though, is the ability for the system to automatically report on the data via email to any relevant parties for a particular student’s attendance. As previously mentioned, lecturers can manage many students at one time, making it difficult for lecturers to individually manage students. With the stresses of university – financing, workload, and balancing their academic work with a job to name a few – having a lack of a watchful eye can compound the problem. Depression and mental health issues have become increasingly recognised as legitimate issues in recent years, and university students dealing with these stress factors as (mostly) young adults makes their emotional well-being an important thing to keep an eye on. A survey of British students held by YouGov in 2016 found that 27% of responders ‘report having a mental health problem of one type or another’, with depression and anxiety the most common at 77% and 74% respectively. If a student begins to lose motivation in their studies, becomes unwell, faces external issues with home life etc., it is likely that their studies and attendance drop in priority. Automatic reporting on a sudden or prolonged absence for individuals can highlight that a student may be struggling, and allows lecturers or studies advisors to proactively reach out to help. The author is also interested in reducing the feedback loop time, as it has been witnessed first-hand through their time in university that collecting formal module feedback only after a module has already been completed provides no benefit to the students who dealt with a poor approach to teaching, and less accountability for lecturers who are expected to meet a university-level standard for teaching.

The author believes, as well as the key stakeholder, that such a solution stands to benefit both lecturers and students: lecturers will be able to manage their attendance feedback more easily and gain some feedback on how any students that they manage are doing; and students can potentially receive help without needing to actively reach out, and can take more ownership in the quality of their teaching and provide more granular feedback as the academic year progresses.

## Project Aims and Objectives

### Aims

The aim of this project is to create a system to process audience data, starting with attendance data and module feedback, to help query, evaluate, and report on such data to any relevant parties. This system will extend and provide new functionality over the existing system.

### Stakeholders

Stakeholders will be informed and managed via regular face-to-face meetings, online meetings, email correspondence, or through other stakeholders.

|  |  |  |
| --- | --- | --- |
| Name / Organisation | Telephone | Email |
| Professor Colin Turner  *Ulster University* | 028 9036 6278 | [c.turner@ulster.ac.uk](mailto:c.turner@ulster.ac.uk) |
| Dr Kenneth Adamson  *Ulster University* | 028 9036 8163 | [k.adamson@ulster.ac.uk](mailto:k.adamson@ulster.ac.uk) |
| Dr Alan Brown  *Ulster University* | 028 9036 8814 | [a.brown@ulster.ac.uk](mailto:a.brown@ulster.ac.uk) |

### Objectives

The system will consist of a web-app with a back-end database to store the data to perform processing on. All lecturers and students will have an account registered for them at the project’s inception, and further updates can be made by administrative system users. Lecturers will be able to view student attendance at different levels of granularity for a module that they lecture – per session, week, and the overall module – and the same for students with their own enrolled modules. Heuristic methods will be applied to the data to provide email reporting to users when applicable. A login page will provide information-hiding so that users only see what’s relevant to them. A relational database will hold the data in suitable models to act on the data in an intelligent manner.

The main project objectives are as follows:

1. Create database with appropriate models to hold audience feedback data as they’re relevant to students and lecturers
2. Create a web-app with consideration for usability and accessibility best practices, and to be responsive for use on different devices e.g. mobile devices
3. Provide functions for student attendance data for students and lecturers to use
4. Provide functions for module feedback data for students and lecturers to use

The activities to be carried out prior to tackling the project to achieve the project objectives:

1. The current process for student attendance gathering, processing and storing must be reviewed to fully understand where the shortcomings are and the improvements to be made. The same is to be done for the module feedback process.
2. Programming languages, environments, and other tools e.g. types of database must be reviewed to determine which are most appropriate for the development of this system
3. Detailed requirements must be gathered from the relevant stakeholders; in this case, Professor Colin Turner
4. The type of software development life cycle most appropriate for this project must be determined
5. The database schema for storing all project data must be determined
6. The data to be stored must be reviewed with consideration for how securely it must be stored, and any encryption / hashing methods should be chosen

The activities to be carried out during the project to achieve the project objectives:

1. A mixture of unit, integration, automated and manual tests must be created / carried out at appropriate intervals to ensure that objectives are met and to minimise regression bugs
2. The system must have extensive validation to minimise bugs and ensure the system will be used as intended
3. The system must be demoed to any relevant stakeholders and usability tested at regular intervals to gain feedback and ensure that the development work is being completed to meet expectations.
4. The system must be created so that the UX lends itself to being easy to understand and use by regular users

The activities to be carried out after the project to achieve project completion:

1. The testing process must be documented to prove that the system should behave as expected
2. Documentation must be created for administrative users of the system for information on how to carry out administrative actions with the system
3. A project report must be created
4. Any relevant hand-off tasks must be completed for future maintenance and usage of the system

## Project Lifecycle

### Potential Lifecycle Models

After evaluating the project, the following information has been gathered:

1. The developer has regularly scheduled meetings and direct communication with the project stakeholders
2. The project is open to constant expansion for different types of audience data
3. There is minimal non-coursework related project documentation to be completed
4. The key stakeholder is very invested in the solution and is open to frequent cooperation regarding the project as it progresses

With this information, the project’s needs can be evaluated against some well-known software development lifecycle models and a choice can be made for whichever fits the project best.

Waterfall

The waterfall model is a distinct set of stages from analysis through requirements gathering to the actual development, testing and ending in documentation and maintenance. Waterfall is straightforward and easy to follow but quite restrictive – there’s little room for overlap in each phase and the developer would only consider it for the project if there was no capacity for constant contact with the end user(s), which isn’t the case.

Spiral

The spiral model differs from waterfall in that it is an iterative model, which is a step in the right direction for the project. The spiral model iterates over 4 steps, the time spent on each growing as the project progresses and grows in complexity. This model places a large amount of focus on prototyping and risk analysis, and while the prototyping aligns with the project goals, the risk analysis is not as important. The spiral model is more appropriate for large projects where mistakes can be costly, and as such it isn’t the right model for this project.

Incremental

A purely incremental approach to software development focuses on building features to full completion, feature by feature. Each solution goes from zero to fully fleshed out, and so an incremental approach is an iterative approach. This type of approach is a viable candidate for the project, but as the development team consists of only a sole developer, this approach could be difficult as features will take longer to fully develop for demoing to stakeholders.

Iterative

A purely iterative approach builds the whole of the project, but to a lower level of capability than the purely incremental approach. This method allows for a barebones-level project to be built, and then refined and extended until each feature is fully completed. This approach is another viable candidate for the project, but just as in the purely incremental approach, with such a small development team capacity it can be difficult to build the whole application to even a basic level as it will take some time, and the stakeholders’ involvement can’t be used as advantageously.

Agile

An agile approach to software development is both incremental and iterative, allowing for the product to be built piece by piece, with each feature gradually improved upon over time. Agile approaches focus on communication with the customer / end user, and continuous delivery of working software. The agile methodology also focuses on flexibility and tailoring each approach to the individual projects that they’re applied to. One of the more popular agile frameworks is Scrum, where demonstrations to the end user or a product owner are frequent (every 2-4 weeks at the end of a ‘sprint’). This aligns nicely with the stakeholders’ availability, and they can also act to keep the project requirements prioritised for the system if the schedule began to slip, which is always a real possibility with software projects. There’s also a minimum need for documentation as the actual software is continually reviewed by the product owner.

### Chosen Lifecycle Model

From the above evaluation, it seems that an agile approach is the best fit for the project. While an agile approach makes more sense for a team of developers and testers rather than a single developer / tester, the methodology is what suits the project’s situation and can be adapted to be more suitable e.g. not placing as much importance on retrospective sessions. The opportunity for the developer to have frequent face to face meetings with the key stakeholder, who will act as the product owner, allows for the ability to constantly demo new work / evaluate and re-prioritise tasks based on progress. The product owner also has ideas for future expansion of the proposed project, so even if the progress made on the project was better than expected the project can still be built upon with (currently) unplanned features.

## Project Deliverables

### Timeline

The total time allocated to this project officially begins on 27th September 2017 and ends on 22nd April 2018.

### Project Deliverables

The strict deadlines for the project are as follows:

|  |  |
| --- | --- |
| Deadline | Date |
| Project Plan | 29th September 2017 |
| ePortfolio | 10th December 2017 |
| Project Monitoring & Control Report | 11th February 2018 |
| Main Project Report & Delivery of Software | 22nd April 2018 |

To define the project scope and schedule / plan tasks, a Work Breakdown Structure (*Appendix 1*) and Gantt Chart (*Appendix 2*) have been created. To accommodate these, the project has been broken down into smaller chunks that are easier to estimate and plan with confidence. The Gantt chart has been scheduled into 2-week development sprints, with 6 man-days per sprint. This value was obtained by estimating a capacity of 3 man-days per week – the sole developer of the project will be working full-time during the project’s lifecycle, and has estimated the capacity to work 2 hours on 4 weekdays and 8 hours on each weekend day. Milestones in the chart have been reserved for the hard deadlines listed above i.e. for coursework piece hand-ins. There is also quite a substantial buffer reserved at the end of the chart for any unexpected difficulties, both technically and non-project related. Any circumstances which take time away from the developer will hold up the entire project, so a safe buffer is in place to mitigate these potential effects. Alternatively, if the project goes according to schedule / ahead of schedule, additional features have been discussed with the key stakeholder informally which could be implemented in this time given the developer has the capacity to do so.

## Project Planning

### Resource Consideration

As this project only plans to deliver a web application, the resources required are minimal:

* Development machine – desktop computer or laptop
* Software to facilitate writing / running / testing code – all free and available on the Internet
* Server to deploy delivered code to

### Risk Assessment

As the project isn’t safety critical and doesn’t deal with very sensitive information, the risk associated is small. Nevertheless, a plan of action should be developed for any potential risks that may occur, including risks that are common to any software project.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Risk | Impact | Controls |
| a | Data loss | Time spent on re-doing work which could be spent on new work – falling behind schedule | Use of version control which also acts as a backup, hosted on a popular site that is unlikely to fail (Github) |
| b | Schedule slippage | Failure to meet critical deadlines and overall project deadline | Early use of and constant referral to the WBS and Gantt chart to manage the project’s scope and time constraints |
| c | Feature creep | Project schedule lengthened (can’t afford this for this project) or quality of work decreases to perform more in less time. Could result in increased costs depending on the new requirements | Overall system requirements are defined and locked down with stakeholders at the start of the project, as well as frequent communication with stakeholders |
| d | Unforeseen circumstances | Unforeseen circumstances, both technical and personal, could lengthen project schedule or decrease quality of work. This project only has a single developer with no replacements, so this could lead to serious setbacks | Accounted for to a certain degree with a substantial buffer at the end of the Gantt chart before the project’s deadline. Beyond a certain point there’s nothing that can be done to fix this, but it should mitigate the risks to a reasonable degree |
| e | Unavailable resources | Project schedule could be lengthened or could even be made impossible depending on the resources required. Workarounds could decrease quality of project | Resources required are identified at the start of the project, and no acceptance of any requirements past this point that change these expected resources |

Below are probability-impact matrices for these risks before and after the proposed controls used to mitigate / prevent their potential impact.

**Before Controls**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Probability** | | | | |
|  |  | 1 | 2 | 3 | 4 | 5 |
| **Impact** | 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  | **c, e** |  |  |
| 4 |  |  | **b** |  |  |
| 5 |  | **a** |  | **d** |  |

**After Controls**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Probability** | | | | |
|  |  | 1 | 2 | 3 | 4 | 5 |
| **Impact** | 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 | **c, e** |  |  |  |  |
| 4 | **b** |  |  |  |  |
| 5 | **a, d** |  |  |  |  |

In the matrices above, the combination of probability and impact of each risk is shown, each falling into a coloured zone. Green represents safety – either the probability or the impact is low enough that the project can be accepted with confidence. Yellow represents a warning – if the risk cannot be reasonably mitigated any further, then it’s still okay to accept the project but it must be kept in mind and monitored. Red represents a problematic risk – the combination of the risk’s probability to occur and the severity of the impact of its occurrence are too problematic to safely accept the project.

As shown, after the controls are applied all of the project risks can be considered safe. Risk ‘d’ – unforeseen circumstances – was initially considered to be the largest risk, but with over a month’s buffer added before the end of the project deadline, this should be reasonably mitigated.

### Data Management

While the project doesn’t deal with very sensitive data, it does handle some personally-identifying information regarding its users. The Data Protection Act 1998, amongst other things, specified that data must be kept safely and securely. With that in mind, any personally-identifiable information will have to be stored and transmitted securely. An appropriately secure encryption method will be used to store user information and the same for hashing user passwords. HTTPS will be used for secure transmission of data, with SSL/TLS certificates supplied and managed by Let’s Encrypt, a free Certificate Authority. As the code is in a public Github repository, any sensitive code-related information e.g. keys will be stored as environment variables on any machine that needs to run the project. In addition to this, no database data will be stored in the repository, only the project code used to process the data, and for the majority of the project’s lifecycle all development will be done on the developer’s local system with obfuscated de-personalised data.

### Knowledge and Skills Required

The system to be developed doesn’t involve the implementation of any hardware components to use, so the only new skills required will be related to actual programming. The project developer has 4 years of experience in the software industry and is familiar with programming and programming concepts, however the proposed project is to be built with Python and Django, which the developer has had no exposure to. There will be some upskilling with this language and framework to be considered as part of the project’s development. The developer is also familiar with databases and there shouldn’t be a need for advanced database functions, so the type of RDBMS used shouldn’t matter much aside from syntactical differences. The developer’s career doesn’t typically involve web-app development so there may be some additional knowledge required in this area, but they’ve had some exposure to this type of development through their university course and personal projects. Any gaps in any of these areas should be sufficiently filled through practice as the project progresses, so no specific planning is needed for upskilling.

### Involving Stakeholders

The key stakeholder, Professor Colin Turner, will also act as the product owner in an Agile setting over the course of the project. Meetings have been scheduled on a weekly / fortnightly basis via Skype or in face-to-face meetings. This will help to ensure that the project is kept on track, and will allow for feature demos and to resolve any queries regarding the system’s requirements or design. Professor Turner also, as the head of the engineering school at the university, will maintain a network of other lecturers who will be likely to use the final system, so he can collect any pertinent additional information from them and pass it directly to the project developer. Dr Kenneth Adamson plays a smaller role in the development of the project, but will be available on a weekly basis for face-to-face meetings regarding any overall project queries and deadlines. Dr Alan Brown will not be directly involved in the project but will communicate with Professor Turner to collate feedback through one end user. Having the stakeholders available so frequently and for face-to-face meetings will allow any hiccups to be resolved quickly and effectively, which will reduce the risk of the project failing to meet its deadlines.

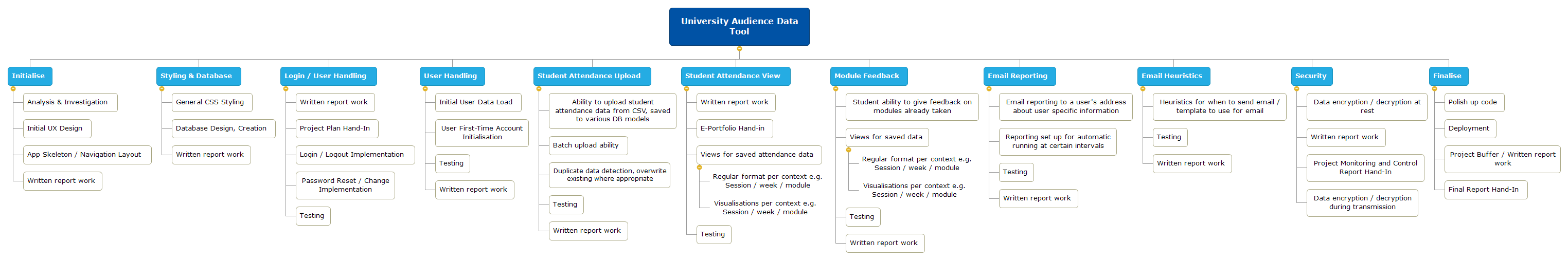
## References

Scotland, K. (2009) *Agile | Fidelity – The Lost Dimension of the Iron Triangle,*Available at: *https://availagility.co.uk/2009/12/22/fidelity-the-lost-dimension-of-the-iron-triangle/*(Accessed: 22nd October 2017).

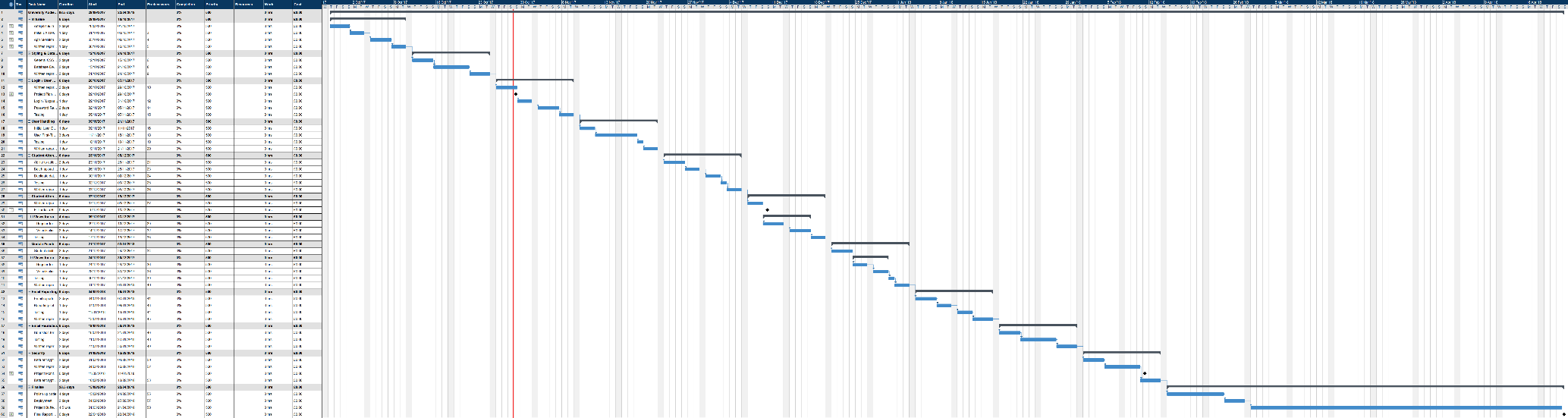
Aronin, S. and Smith, M. (2016) *One in four students suffer from mental health problems,*Available at: *https://yougov.co.uk/news/2016/08/09/quarter-britains-students-are-afflicted-mental-hea/* (Accessed: 24th October 2017).

GOV.UK (n.d.) *Data protection,*Available at: *https://www.gov.uk/data-protection*(Accessed: 24th October 2017).

## Appendix 1



## Appendix 2



## Appendices Files

Appendix 1



Appendix 2



MindView Project

