# Documentation

Extensions and Deferrals Management Website

# Table of Content

[Documentation 1](#_Toc16999006)

[Table of Content 2](#_Toc914632036)

[Connection Details 2](#_Toc1223761517)

[Project Description 2](#_Toc76763893)

[Team Members 3](#_Toc376729026)

[Team Roles 3](#_Toc1335946863)

[Goals for the Project 3](#_Toc1928125630)

[Methodology 4](#_Toc1291493141)

[Requirements 4](#_Toc1064512575)

[Non-functional Requirements 5](#_Toc1219362229)

[Estimated Project Cost 6](#_Toc1328586766)

[Risk Assessment 6](#_Toc991776693)

[Test Plan 10](#_Toc2137973385)

[Initial Inquiries to Project Client 12](#_Toc1147841304)

[UX Laws That Impact Design Decisions 14](#_Toc1969656548)

[Accessibility 17](#_Toc2141086142)

[NOTES 18](#_Toc1807875712)

[Links for Collaborative Work 19](#_Toc576008455)

[Reference List 19](#_Toc1047741650)

## Connection Details

Username: [admin@email.com](mailto:admin@email.com) Password: P@55word

Username: [member@test.com](mailto:member@test.com) Password: P@55word

Database name: db\_2225624\_latework

Database username: user\_db\_2225624\_latework

Password: P@55word

Web app link: [https://2225624-co5019.win.studentwebserver.co.uk/CO5019/Index](https://teams.microsoft.com/l/message/19:4c207639-0408-4e20-a5f9-300cfb855b0f_71ba2512-53ec-424e-a18d-036d5e0ab35a@unq.gbl.spaces/1715861934395?context=%7B%22contextType%22%3A%22chat%22%7D)

Miro: [Late Mark Database, Visual Workspace for Innovation (miro.com)](https://miro.com/app/board/uXjVKRSQROc=/)

Trello: [Late Penalties Project (latepenaltiesproject) | Trello](https://trello.com/w/latepenaltiesproject)

GitHub repository link: <https://github.com/AdrianSavaUoC/TheFoodJoint>

## Project Description

This project involves the development of a Database System to monitor extensions, late penalty waivers, and deferral requests from students enrolled at the School of Computer and Engineering Sciences and partner colleges. This will be used by the client, Richard Stocker. The design of this system aimed to achieve a smooth integration with the Chester Network, providing a significant improvement compared to the preceding Excel-based system.  
  
The database incorporates features that facilitate the management of input pertaining to student requests for extensions, deferrals, and other relevant information. By enabling advanced searches based on criteria including Student Number, Module, Course, and Extension Type, it facilitates the retrieval of data in a targeted and efficient manner, which contributes to improved reporting and decision-making.

## Team Members

Adrian Sava - 2225624

Jason Nicholls - 2205497

Alex Adams – 2106870

Alexia Okeregbe - 2215193

Owen Kopp - 2111783

Mai Anh Cao - 2209115

## Team Roles

Project Manager/Team Leader- Adrian

Communication Lead-Adrian

UI/UX and Design- Mai Anh and Jason

Documentation- Alexia, Alex, Adrian, Jason, Owen, Mai Anh

Testing- Alexia, Owen, Mai Anh

Developers –Adrian, Jason, Alex, Owen, Mai Anh, Alexia

## Goals for the Project

The goals for this project were to replace the current University of Chester late penalty system with the one that we have created, this is because with the previous system, to calculate late penalties, the user has to have multiple excel spreadsheets open and cross reference them manually and find out if student had an extension, if the extension request was accepted, how many days the assignment was handed in late, if any and then from this calculate the number of late marks the student should receive. This is done much easier by having the data all in the same database and the user can now just search for the student, course or module and see there in the database how many late marks they should be given. The new system now has data consistent across the database as before there were multiple different ways to input the same data, which can make searching difficult, we have added dropdowns where necessary for inputting data, only allowing set options such as “Approved” and “Not Approved” to be in certain columns, instead of the user being able to type any kind of valid answer.

## Methodology

The methodology that was chosen for this project is the Agile methodology. This is due to its user-centric nature (Fox et al., 2008), execution of tasks in sprints, and how these features are a good fit for a project of this nature. This methodology prevents more errors after deployment, ensures that the system satisfies users’ needs, and creates room for more testing and revision of functional requirements before deployment.

As opposed to other traditional methodologies like waterfall (Sommerville, 2018), Agile methodologies allow changes to be made to parts of the system after development was already completed. For example, if the client decided they wanted to add a new or different functionality to the original design or if in the early stages features that were not needed or not what the client wanted due to a misunderstanding were included in a design, we could easily change these aspects of the system before moving onto other parts of the project. This can also be done in tandem with the other parts of the project.

## Requirements

Some of the functional requirements for the project are as follows:

* **The database should be searchable** i.e., there should be a search box or dropdown menus that allow the user to narrow down the data that they are presented. The dropdown menus should be used to search by:
  + Student Number
  + Name
  + Module
  + Course
  + Year
  + Location
  + Extension Type
  + Expected Submission Date
  + Actual Submission Date
* Must be able to be used every year
* If there are no search criteria input, all records (e.g., SELECT ALL) should be displayed by default
* The user must be able to input data into the database, using dropdown menus and input boxes where appropriate to do so, input criteria include:
  + Student Number
  + Name
  + Module
  + Course
  + Year
  + Location
  + Extension Type
  + Late Penalty Waiver
  + Evidence
  + Approval or Denial
  + Expected Submission Date
  + Actual Submission Date
* This project will be implemented into the existing chester.network page, and fit its theme and layout
* Accessibility
* Export feature to PDF and/or Excel

Nice to have features for the project include:

* Summary tiles that use the “drag and drop” process so they automatically fill in details to filter the database or search for.

## Non-functional Requirements

* Scalability: Scalability is the ability of a website to perform with variation in load or traffic quality. The website needs to be highly scalable so it can be used effectively in high traffic periods as the project aims to ensure that the client and users will be able to access the data from all cohorts input extensions and export the database to a pdf/excel document within the program up to standard.
* Usability: Users should be able to use the website effectively and efficiently as the requirement is crucial as the previous system is considered confusing to the client which creates a requirement that the new system will need to be approachable. This because the client states that multiple spreadsheets can be confusing, and the number of tabs can cause users to miss information which impacts the intended service of the application so the usability aspects must be considered in development.
* Performance: This is the speed at which users’ queries are responded to. The website needs high performance so that users’ goals are met efficiently and reduce pain points. The criteria will be met as the data must be accessible to the client and the functions involving calculating the results and exporting the data must be efficient and accessible to the client.
* Security: Users’ data and identities should be protected as they use the site, to meet this requirement, a login system will be created to ensure that the client’s application is secure and only accessible to users that have permission to prevent potential damage to the database or system misuse.
* Availability: Users should be able to always use the website regarding the performance on the user’s system and availability on multiple devices such as computer’s phones and tablets along with being compatible with resolution sizes so all information are displayed intended and in a simple format for the client.

## Estimated Project Cost

To determine the total cost for the project management task, multiple factors must be considered such as the hourly rate of pay, the amount of hours of work required each day, and potential absences that may occur during the work period due to unexpected emergencies/limitations or health issues.

According to (Junior Developer Salary in United Kingdom - Average Salary, 2024) we determined that the median hourly rate for a junior developer is £14.70 in the UK. Next, we calculated the number of hours worked on this project, each team member worked from 09:00 to 17:30, 5 days a week for 5 weeks, which gives 212.5 hours of total work, multiplied by the determined hourly rate gives £3123.75 total per person. As there are 6 members on the team, the total cost for the project is £18,742.50.

This figure doesn’t include any business expenses such as equipment or software licenses, it also doesn’t include potential additional costs for maintenance of the system. It is only a rough estimate for the whole project and not a definitive value.

## Risk Assessment

**Key:**

Likelihood is the measure of how likely the risk is to occur

Impact is the measure of if the risk were to occur, how big of a problem it would be for the project

Risk Rating is calculated by multiplying the likelihood and impact, and is used to measure the overall severity of the risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood (1-5)  Low to high risk | Impact (1-5)  Low to high risk | Risk Rating (Likelihood \* Impact) | Mitigation Techniques |
| Data Breach | 2 | 5 | 10 | Techniques used by attackers for data breach would be identified and ways for combatting them would be implemented in development of the website.  For example, Input validation would be added to the website to prevent SQL Injection |
| Loss of data integrity | 2 | 4 | 8 | Data in the website would be backed up and input validation and error handling would be implemented in development of the website |
| Loss of Code Integrity | 2 | 4 | 8 | Version Control will be implemented in the project with Gitlab.  Effective communication between team members would be prioritised to make sure that unwanted changes are not made to code. |
| Non-compliance with standards | 1 | 4 | 4 | Examine requirements of standards such as GDPR, WCAG, etc. And ensure that requirements are met.  Research will be carried out on existing web standards such as GDPR and WCAG and measures will be put in place to ensure the requirements of these standards are met. |
| System Failure | 2 | 4 | 8 | Adequate testing will be carried out on the site. Some types of testing that would be carried out include integration testing, Unit testing and User testing with the client and other stakeholders.  Version control would also be put in place. |
| Under estimation and over estimation of project requirements | 3 | 3 | 9 | Proper planning and time management, implementation of sprints to keep track of changes in requirements.  The sprint featured in the Agile Methodology which has been chosen for this project would ensure that there is constant monitoring of project goals and revision of project requirements.  Work management tools like Trello and Miro would also be used to keep track of progress in the project. |
| Poor communication with Clients | 2 | 4 | 8 | Regular meetings will be held with Clients to ensure that they are carried along at every stage of development. The client will also be highly involved in user testing to make sure the finished product satisfies their needs. |
| Lack of Resources for project requirements | 1 | 4 | 4 | There would be continuous examination of project requirements and discussions with the Stakeholders if the requirements exceed available resources. |
| Inability of team members to perform as estimated | 2 | 4 | 8 | There would be regular and proper communication between members of the team to know how much they can contribute.  Possible reasons for inability to work such as sickness would be examined and discussed with team members.  Any instance of this would be reported to mentors and tutors for them to provide support in this case. |

## Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Description | Inputted Data | Expected Outcome | Actual Outcome | What fixes were required? |
| 1 | Does the database appear on the ASP Webpage |  | To be able to access the database that's hosted on the ASP webpage. |  |  |
| 2 | Can the user input data into the table |  | User can input data into an empty table that fits the required data. |  |  |
| 3 | Can the user edit data that already exists in the database |  | To make a change to existing data if necessary (example: changing the approval of extension) |  |  |
| 4 | Does the Search function load the required data. |  | To display data that only specific to what was searched. (example: module code shows all data for the single module) |  |  |
| 5 | Can the program export data to a PDF or Excel document which the user can access. |  | To allow the user to download a PDF or excel document off from the website and onto their system. |  |  |
| 6 | Can the user filter data based on multiple criteria? |  | It should be possible for the user to apply multiple filters at once (by category, status, date range, etc.) to customise the displayed data to suit their needs. After applying the filters, the filtered data should update instantly, and the user should have no trouble clearing the filters. |  |  |

A more detailed Test plan with results can be found in the testing document in the project portfolio.

## Initial Inquiries to Project Client

During the development period, client meetings were be held to allow the project group to present Inquiries on design concerns and UI intractability. This allowed both the client and the project team to have a further understanding how the project will be developed and to understand the overall vision and requirements that the client wanted for the project.

Week 1 Project Questions and feedback: 28/03/2024

Question: Will the database be stored on a website

Response: Yes, the database will need to be formatted into a website to store late penalties which admins can add.

Question: what programming language is ideal for the project?

Response: Any programming language should be good, whatever programming language you are familiar with should be ideal.

Question: Should the project team remake the CSV?

Response: It's not required, the focus should be the database.

Question: what about the self-certification in database, how should it be approached?

Response: with self-certification, it does not get automatically approved so it good to attach documents/evidence to the database.

Question: Do we need to make a login system

Response: No, this application will be implemented to the university system.

Question: Should the database be displayed on a single table

Response: It's up to the project team to decide, determine how to design this system (client not considered this part)

Question: in terms of meeting what time would be suitable?

Response: Any day that's not Tuesday should be ideal, sending emails of inquiries and meetings is fine as the client is located close on site and accessible.

Week 3 Project Questions and feedback 29/04/2024, 10:30am

* Is it required for the database project to be functional on all devices? (PC, Mobile devices, etc)
* Should there be a level of permission or login to the user which grants or rejects deferrals?
* How will the relation work between the user that inputs the data of the late penalties and the main database that determines and extension, deferral, or rejection.
* Is the current design wireframe ideal to the result that is required, are there any changes and is there any missing features or desirable additions to the system?

Week 4 Project Questions and feedback: 10/05/2024

Client: On the extensions table, are there any filters for the data provided.

Team Response: We are currently still working on implementing this feature.

* Looking like the FSE network which is nice
* Not sure what to suggest but when the filtering is implemented, the website should be nice, but the design is overall good.
* Good that you can check the student information
* It might be good in the assignment if it tells the user how many students have applied to the extensions, mainly that have been accepted
* You would not want to delete any data, if it goes live, its not ideal as student data is important
* Overall, the system is well put together, and the design resembles the Chester FSE website nicely.

## UX Laws That Impact Design Decisions

To ensure that that the design decisions that are considered during the design phase of the project are valid and necessary, following the laws of UX will be necessary to the specific requirements that are needed to complete the project. The decision factor for the UX laws will involve the client’s vision of the project and the requirements that they presented in both the scenario and meetings, Yablonski (2024) presents varying UX laws that are considered as the fundamentals of creating a project that involves the intractability, simplicity and the overall general display of information in the database.

**Aesthetic-Usability Effect:** is a UX law that focuses on creating a ideal visual layout that has the right aesthetic for the client, this law is ideal as the client finds the original program confusing and simplicity in a program is important.

The Aesthetic-Usability Effect will be important to the project because it aims to simplify the design of the database program and to implement more functionality features (e.g. exporting data as pdf) towards the client and users as it improves general efficiency of the programs purpose.

**Fitts’s Law:** Fitt’s law is the relationship between how fast a person can see an element on the screen and then how accurately and quickly they can navigate to it. When designing the UI for our database we should make buttons and other elements the user interacts with big enough to be accurately clicked and spaced apart enough as to not accidentally click the wrong option.

Bakaev & Razumnikova, (2021) presents that the speed and interaction of data interaction between the user and source of the UI will be crucial to meet the criteria of Fitt’s law as its optimal that the data is interactable immediately when displayed on the online UI.

Additionally, Fitt’s Law will support the final program as it improves the visibility and usability of the program, this means that it will create a user-friendly interface that is simplified as the client provides that manually interacting with the program can be confusing so Fitt’s Law should be the solution that aims to improve intractability with the program.

**Hick’s Law:** will focus on ensuring that the project is simplified on use for the client, this will involve features that simplify tasks for the client and users. Bakaev & Razumnikova, (2021) additionally express how the Hick’s law is a factor that can ensure that the overall development of the program follows the interaction and to always evaluate the general layout so its not overly complex for the client. For example, the database program may have a search feature which allows the user to find specific students or those in a specific module more efficiently as the feature saves time. Additionally, it is crucial that the project stays to its intended goal as it must always perform the intended requirements as simplifying too much may make the program less ideal if it does not perform natural functions.

Hick’s Law was utilised to improve the user approach of the application by simplifying the steps it takes to perform the required activity (example: changing data for mark deferral) so that its efficient and user friendly to the users that will use the application.

**Law of proximity:** Prioritises the general layout of the objects in the user interface, this design aspect will involve how the database will appear and how objects such as buttons will need to be placed correctly to ensure the project meets the client’s requirement and to provide efficiency.

For the project program, the law of proximity was used when developing the UI of the program as improving the data layout can simplify the program for the client to use.

**Parkinson’s Law:** presents itself as an approach to measure individual tasks in a specific duration to ensure that the user can perform the intended productive tasks within a limited time, this is important to the client as the development goal of the program is to create a simplified version of the late penalties database and additional features such as a button creating a excel/pdf report can fulfil the requirements to the client.

Parkinson’s Law was applied to the project as the target aim of the final product is to create a simple interface which improves the accessibility of changing and viewing data that's on the application.

**Miller’s Law:** Miller’s Law is an approach that follows the concept that every individual person on average can memorise and focus up to around 7 different chunks of information at a time. The general rule of the following law is to ensure that the program will not be overwhelming and to consider the different users that may use the application as the user’s memory can vary regardless of the medium value as displaying information and providing functionality in the main development priority.

Miller’s Law was used to ensure that the program UI will not be overwhelming for users and to implement a search feature for the code so that only the specific selection of data will be shown which should overall allow the user to focus on the selected data that they require while performing the task.

**Tesler’s Law** is a project law that covers the concept that all forms of process may have a form of complexity that is limited when it comes to being simplified. To solve this, the project can be built to ensure that the functionality is simple and approachable to the client and users, this can be done by labelling or using a format to ensure that information is accessible to the user.

To ensure that the project application meets the criteria of the Tesler’s Law, the functionality and UI of the application was designed to have usability that allows any staff member of any technical skills to utilises the features provided without difficulty.

## Accessibility

**Accessibility** - Making sure that all users could access and use the application efficiently, regardless of their abilities, was the main goal when designing for accessibility (Initiative, 2005). These design choices and the rationale behind them, derived from standards and guidelines like the Laws of UX and the Web Content Accessibility Guidelines (WCAG), are as follows:

1. **Colour Contrast**:

* ***Guideline 1.4 – Distinguishable****:*
* ***Principle****: Level AA, ensuring the visual presentation of text and images of text has a contrast ratio of at least 4.5:1.*

1. **Keyboard Accessibility:**

* ***Guideline 2.1 – Keyboard Accessible:***
* ***Principle****: Level A, ensuring all functionality is operable through a keyboard without specific timings for keystrokes.*

1. **Clear and Consistent Navigation**

* **Guideline 2.4 – Navigable:**
* **Principle:** Level A and AA, focusing on bypassing blocks of content and keeping consistent navigation.

1. **Readable Font Sizes and Styles:**

* ***Guideline 1.4 – Distinguishable:***
* ***Principle:*** *Level AA, allowing text to be resized up to 200% without loss of content or functionality.*

1. **Descriptive Alt Text for Images:**

* **Guideline 1.1 – Text Alternatives:**
* **Principle:** Level A, providing text alternatives for any non-text content so it can be changed into other accessible forms.

1. **Accessible Forms:**

* ***Guideline 1.3 – Adaptable:***
* **Principle:** Level A, ensuring information, structure, and relationships are conveyed in a programmatically accessible manner.

1. **Feedback and Error Handling:**

* **Guideline 3.3 – Input Assistance:**
* **Principle:** Level A and AA, identifying and describing input errors and providing suggestions for correction.

1. **Inclusive Language and Design:**

* ***Guideline 3.1 – Readable:***
* *Principle: Level A, providing clear language and communication throughout the interface for ease of understanding.*

## NOTES

**Level A:**

\* **Definition:** *Level A represents the lowest level of accessibility, focusing on the fundamental and indispensable elements required for web accessibility.*

*\** **Requirement:** *To achieve at Level A, a webpage must satisfy all Level A criteria. This encompasses the provision of fundamental textual alternatives for non-textual content, guaranteeing accessibility through keyboard input, and refraining from including any content that may induce seizures.*

**Level AA:**

\* ***Definition:*** *Level AA enhances accessibility beyond Level A by incorporating additional requirements to cater to a wider range of users.*

*\** **Requirement:** *To achieve Level AA compliance, a webpage must satisfy the criteria for both Level A and Level AA. This encompasses stricter criteria for text contrast, guaranteeing the ability to resize text, and offering captions for live audio content.*

**Level AAA:**

**\* Definition:** *Level AAA refers to the highest level of accessibility, which aims to offer the optimal experience for all users, including those with significant disabilities.*

*\** **Requirement:** *To achieve Level AAA, a webpage must fulfil the criteria for Level A, Level AA, and Level AAA. This encompasses more stringent contrast criteria, supplementary audio narration for pre-recorded video material, and the provision of expanded explanations for media.*

## Future Improvements

Some features which could be added on the next Software Development Life Cycle are listed below:

* Graphs and Charts for reports added to the export feature
* Export for the details page
* Student Name column added to Deferrals, Extensions, and Late Penalty Waivers Tables.
* Better placement for the tables in the pdf files from the “Export to PDF” feature.’

## Links for Collaborative Work

**(moved the links here too so that everything can be in 1 place)**

Miro: [Late Mark Database, Visual Workspace for Innovation (miro.com)](https://miro.com/app/board/uXjVKRSQROc=/)

Trello: [Late Penalties Project (latepenaltiesproject) | Trello](https://trello.com/w/latepenaltiesproject)

Gitlab: [JASON NICHOLLS / ExtensionDeferrals · GitLab (chester.network)](https://git.chester.network/2205497/extensiondeferrals)

## Reference List

Yablonski, J. (2024). Laws of UX. Retrieved from https://lawsofux.com/

Bakaev, M. & Razumnikova, O. (2021) What Makes a UI Simple? Difficulty and Complexity in Tasks Engaging Visual-Spatial Working Memory, 13(1), 2-16.

Fox, D., Sillito, J., & Maurer, F. (2008). Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry. *Agile 2008 Conference*. <https://doi.org/10.1109/agile.2008.78>

*J*unior Developer salary in United Kingdom - Average salary. (2024). Talent.com. <https://uk.talent.com/salary?job=junior+developer>

*How to Meet WCAG (Quickref Reference)*. (2023, November 12). Retrieved April 24, 2024, from <https://www.w3.org/WAI/WCAG22/quickref/>

Initiative, W. W. A. (2005, July). *WCAG 2 Overview*. Web Accessibility Initiative (WAI). Retrieved May 1, 2024, from <https://www.w3.org/WAI/standards-guidelines/wcag/>

Sommerville, I. (2018). *Software Engineering*. Pearson.