

Eagna

Interim Report

TU856

BSc in Computer Science

**Conor Davis**

**C20441826**

**Supervisor**  
Eoin Rogers

School of Computer Science

Technological University, Dublin

**29 / 10 / 2025**

Abstract

This project was designed as a solution to the issues faced in registration and usage of Brightspace during the Academic Year of 2025 / 2026.

Through deep research and meticulous design, this web application, at the centre of a distributed system, containing three separate machines for the web application, database and monitoring system, is an approach to creating a condensed, functional portal to support registration and login for students, to access their modules and the necessary content to engage in their studies. It also provides login access for lecturers to interact with their modules in the way they need, along with an administration panel for maintenance by staff.

[GitHub Repository](https://github.com/cdavis471/Eagna-FYP-TU856)

Declaration

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

Signed:

Conor Davis

Conor Davis

29 / 10 / 2025

Acknowledgements

I would like to thank everyone that has helped me in my studies at the Technological University of Dublin.

I would like to thank Eoin Rogers, my Final Year Project supervisor, for his help throughout the creation of this project.

I would like to thank the students and lecturers who took the time to fill out the questionnaires I created for my Requirements Gathering.

I would like to thank Damian Bourke & Mariana Rocha for the immense support they provided through some of the roughest moments during my time at TUD.

I would also like to thank Patricia O’Byrne for her support at the end of my third year, before her retirement.

Contents

[1. Introduction 1](#_Toc214700357)

[1.1 Project Background 1](#_Toc214700358)

[1.2 Project Description 1](#_Toc214700359)

[1.3 Project Aims and Objectives 2](#_Toc214700360)

[1.4 Project Scope 6](#_Toc214700361)

[1.5 Thesis Roadmap 6](#_Toc214700362)

[2. Literature Review 7](#_Toc214700363)

[2.1 Introduction 7](#_Toc214700364)

[2.2 Alternative Existing Solutions 7](#_Toc214700365)

[2.3 Core Technologies 12](#_Toc214700366)

[2.4 Requirements 21](#_Toc214700367)

[2.5 Existing Final Year Projects 27](#_Toc214700368)

[2.6 Overall Conclusion 29](#_Toc214700369)

[3. System Analysis 30](#_Toc214700370)

[3.1 System Overview 30](#_Toc214700371)

[3.2 Requirements Outline 30](#_Toc214700372)

[3.3 Questionnaires 31](#_Toc214700373)

[3.4 Questionnaire Feedback 34](#_Toc214700374)

[3.5 Brightspace Features Analysis 45](#_Toc214700375)

[3.6 Nielsen’s Heuristics 49](#_Toc214700376)

[3.7 Requirements Analysis 50](#_Toc214700377)

[3.8 Initial System Specification 55](#_Toc214700378)

[3.9 Conclusions 58](#_Toc214700379)

[4. System Design 59](#_Toc214700380)

[4.1 Introduction 59](#_Toc214700381)

[4.2 Software Methodology 59](#_Toc214700382)

[4.3 The Overall Model 62](#_Toc214700383)

[4.4 The Features List 63](#_Toc214700384)

[4.5 Plan By Feature 65](#_Toc214700385)

[4.6 Design By Feature 67](#_Toc214700386)

[4.7 Conclusions 72](#_Toc214700387)

[5. Testing and Evaluation 73](#_Toc214700388)

[5.1 Introduction 73](#_Toc214700389)

[5.2 Plan for Testing 73](#_Toc214700390)

[5.3 Plan for Evaluation 78](#_Toc214700391)

[5.4 Conclusions 78](#_Toc214700392)

[6. System Prototype 79](#_Toc214700393)

[6.1 Introduction 79](#_Toc214700394)

[6.2 Prototype Development 79](#_Toc214700395)

[6.3 Results 79](#_Toc214700396)

[6.5 Conclusions 80](#_Toc214700397)

[7. Issues and Future Work 81](#_Toc214700398)

[7.1 Introduction 81](#_Toc214700399)

[7.2 Issues and Risks 81](#_Toc214700400)

[7.3 Plans and Future Work 81](#_Toc214700401)

[7.3.1 Project Plan with GANTT Chart 82](#_Toc214700402)

[8. Appendices 83](#_Toc214700403)

[Appendix A: Potential Technologies Research 83](#_Toc214700404)

[Appendix B: Questionnaire Feedback 86](#_Toc214700405)

[Appendix C: System Model and Analysis 1](#_Toc214700406)

[Appendix D: Design 1](#_Toc214700407)

[Appendix E: Prompts Used with ChatGPT 1](#_Toc214700408)

[Appendix F: Additional Code Samples 1](#_Toc214700409)

[9. References 2](#_Toc214700410)

Table of Figures

[Figure 1 - General Overview of the Eagna System 2](#_Toc214636581)

[Figure 2 - Student Feedback Questionnaire - Which Virtual Learning Environment(s) have you used before? 37](#_Toc214636582)

[Figure 3 - Student Feedback Questionnaire - General Experience Rating 38](#_Toc214636583)

[Figure 4 - Student Feedback Questionnaire - Easiest Feature Word Cloud 38](#_Toc214636584)

[Figure 5 - Student Feedback Questionnaire - Easiest Feature Feedback I 38](#_Toc214636585)

[Figure 6 - Student Feedback Questionnaire - Easiest Feature Feedback II 39](#_Toc214636586)

[Figure 7 - Student Feedback Questionnaire - Easiest Feature Feedback III 39](#_Toc214636587)

[Figure 8 - Student Feedback Questionnaire - Hardest Feature Word Cloud 39](#_Toc214636588)

[Figure 9 - Student Feedback Questionnaire - Hardest Feature Feedback I 39](#_Toc214636589)

[Figure 10 - Student Feedback Questionnaire - Hardest Feature Feedback II 39](#_Toc214636590)

[Figure 11 - Student Feedback Questionnaire - Hardest Feature Feedback III 39](#_Toc214636591)

[Figure 12 - Student Feedback Questionnaire - Hardest Feature Feedback IV 40](#_Toc214636592)

[Figure 13 - Student Feedback Questionnaire - Hardest Feature Feedback V 40](#_Toc214636593)

[Figure 14 - Student Feedback Questionnaire - Easiest Feature Feedback IV 40](#_Toc214636594)

[Figure 15 - Student Feedback Questionnaire - Easiest Feature Feedback V 40](#_Toc214636595)

[Figure 16 - Student Feedback Questionnaire - Easiest Feature Feedback VI 40](#_Toc214636596)

[Figure 17 - Student Feedback Questionnaire - Hardest Feature Feedback VI 41](#_Toc214636597)

[Figure 18 - Student Feedback Questionnaire - Hardest Feature Feedback VII 41](#_Toc214636598)

[Figure 19 - Student Feedback Questionnaire - Hardest Feature Feedback VIII 41](#_Toc214636599)

[Figure 20 - Student Feedback Questionnaire - Hardest Feature Feedback IX 41](#_Toc214636600)

[Figure 21 - Student Feedback Questionnaire - Hardest Feature Feedback X 41](#_Toc214636601)

[Figure 22 - Student Feedback Questionnaire - Hardest Feature Feedback XI 41](#_Toc214636602)

[Figure 23 - Student Feedback Questionnaire - One Thing To Change Word Cloud 42](#_Toc214636603)

[Figure 24 - Student Feedback Questionnaire - Do you identify with any of the following disability categories? 42](#_Toc214636604)

[Figure 25 - Student Feedback Questionnaire - Disability Affected Usage Feedback I 43](#_Toc214636605)

[Figure 26 - Student Feedback Questionnaire - Disability Affected Usage Feedback II 43](#_Toc214636606)

[Figure 27 - Student Feedback Questionnaire - Disability Affected Usage Feedback III 43](#_Toc214636607)

[Figure 28 - Student Feedback Questionnaire - Disability Affected Usage Feedback IV 43](#_Toc214636608)

[Figure 29 - Student Feedback Questionnaire - Suggested Missing Feature I 43](#_Toc214636609)

[Figure 30 - Student Feedback Questionnaire - Suggested Missing Feature II 43](#_Toc214636610)

[Figure 31 - Student Feedback Questionnaire - Suggested Missing Feature III 43](#_Toc214636611)

[Figure 32 - Student Feedback Questionnaire - Suggested Missing Feature IV 43](#_Toc214636612)

[Figure 33 - Student Feedback Questionnaire - Suggested Missing Feature V 43](#_Toc214636613)

[Figure 34 - Student Feedback Questionnaire - Suggested Missing Feature VI 44](#_Toc214636614)

[Figure 35 - Lecturer Feedback Questionnaire - Which Virtual Learning Environment(s) have you used before? 44](#_Toc214636615)

[Figure 36 - Lecturer Feedback Questionnaire - General Experience Rating 45](#_Toc214636616)

[Figure 37 - Lecturer Feedback Questionnaire - Easiest Feature Feedback 45](#_Toc214636617)

[Figure 38 - Lecturer Feedback Questionnaire - Hardest Feature Feedback 45](#_Toc214636618)

[Figure 39 - Lecturer Feedback Questionnaire - One Thing To Change Feedback 46](#_Toc214636619)

[Figure 40 - Lecturer Feedback Questionnaire - Associated Disability Category 46](#_Toc214636620)

[Figure 41 - Lecturer Feedback Questionnaire - How Associated Disability Category Affects VLE Usage 47](#_Toc214636621)

[Figure 42 - Lecturer Feedback Questionnaire - One Missing Feature to Help Deal with Disability Category 47](#_Toc214636622)

[Figure 43 - Brightspace Homepage Header 48](#_Toc214636623)

[Figure 44 - Brightspace Work To Do Widget 48](#_Toc214636624)

[Figure 45 - Brightspace Modules Dashboard 49](#_Toc214636625)

[Figure 46 - Brightspace Module Header & Navigation Menu 49](#_Toc214636626)

[Figure 47 - Brightspace Module Learning Materials & Announcements 50](#_Toc214636627)

[Figure 48 - Brightspace Progress Menu 50](#_Toc214636628)

[Figure 49 - Brightspace Discover Module Feature 51](#_Toc214636629)

[Figure 50 - Initial Student Use Case 54](#_Toc214636630)

[Figure 51 - Initial Lecturer Use Case 55](#_Toc214636631)

[Figure 52 - Expanded Use Case 56](#_Toc214636632)

[Figure 53 - Eagna System Deployment 65](#_Toc214636633)

# 1. Introduction

## 1.1 Project Background

“Users spend most of their time on other websites, so they expect your site to work like all the other sites they already know. When a design deviates from users’ expectations, usability suffers. Don’t be arrogant and assume that your new design idea is so brilliant that it can overrule decades of user habituation.” – Jakob Nielsen [1]

Having created the Law of the Internet User Experience as far back as 2000 [2], Jakob Nielsen has tried to tell designers, for decades, that they must relinquish control of creating their vision of the perfect website, in order to accommodate the novice user who will not have the time of day to learn their website inside out in order to simply begin using it properly.

With the problems faced in the summer by the Technological University of Dublin, regarding the newly created registration system, and the subsequent knock-on effect on Brightspace, the idea to create a condensed, user friendly, and accessible solution to the issue at hand was formed.

This background forms the aim of this project idea. A clean and functional Virtual Learning Environment, tailored to the implicit needs of the students for learning, with a focus on a clear and accessible structured design, with an integrated registration system, supported by the already existing email system, with the provided tools necessary for lecturers to run their modules.

With a secondary aim of removing bloat, while ensuring everything that is required to function is there (including the functionality necessary for lecturers to interact with their modules), this project aims to create a potential in-house solution for the university.

## 1.2 Project Description

Eagna, the Irish word for ‘Wisdom’, is a web application at the centre of a distributed system, designed for the Technological University of Dublin, that would replace the core Brightspace & Registration System that has been put in place. With a single online portal, combining the functionality of a Virtual Learning Environment with an integrated registration system, that will handle the mandatory and optional modules of the courses available, this application would provide the access to the necessary tools and functionality needed to engage in studying at TUD, in an accessible, available and secure manner.

The core design of this system is as follows: a web application, supporting registration and a login, access to course modules, material and assignments, for both student and lecturer. A database will support the functionality of this system, containing fabricated student data to mimic the current registry of students and lecturers. Alongside this, will be a platform monitoring system, to visualize metrics and uptime for system administrators that would be required to keep the system running, should issues arise.

The distributed system will be designed to deploy on either an on-site local network, that could reach the internet, once accommodated by network security, or deployed with a Cloud-as-a-Service (CaaS) provider, should it be preferable to host the system off-site.

A diagram of a web application

AI-generated content may be incorrect.

Figure 1 - General Overview of the Eagna System

## 1.3 Project Aims and Objectives

The overall aim for this project is to deliver a functional and clean Virtual Learning Environment, with an integrated Registration System, that has custom user data associated with it, allowing for access to the application. Here are the following final objectives for this project to be met:

* **Research**
  + Provide a detailed review and analysis with references on the following sections, to have a comprehensive overview of what is best suited for this project:  
    - Conduct a review of relevant contemporary Virtual Learning Environments:  
      * ***Brightspace***
      * ***Moodle***
      * ***Loop***
      * ***Blackboard***
      * ***Canvas***
    - Engage in a literature-based review on relevant core technologies:  
      * ***Web Application Technologies***
        + Front-End (HTML, CSS, HTMX, JavaScript, React)
        + Back-End (Django, Flask, Ruby on Rails)
      * ***Database Technologies***
        + SQL (PostgreSQL, MySQL)
        + NoSQL (MongoDB, Apache Cassandra)
      * ***Monitoring Technologies***
        + Prometheus, Grafana, Zabbix, Datadog
      * ***Virtualization Technologies***
        + Oracle VirtualBox, VMWare
    - Engage in a short review on potential future technologies:  
      *Moved To Appendix A*  
      * ***Caching***
        + Redis, Valkey, Dragonfly
      * ***Container Orchestration Technologies***
        + Kubernetes, ECS Fargate, Docker
      * ***Cloud Services***
        + Amazon AWS, Microsoft Azure & Google Cloud
    - Engage in a short review on requirements:  
      * ***Availability***
      * ***Performance***
      * ***Security***
      * ***General Data Protection Regulation***
      * ***Accessibility***
    - Review previous Final Year Projects to help in gauging standards.  
      * ***Suaimhneas***
      * ***eLearning System For Anonymous Feedback***
* **Requirements Gathering**
  + Understand & collect the necessary requirements for users and stakeholders.
  + Collect and critically review feedback from potential end users that have previously used Virtual Learning Environments.
  + Review the features of Brightspace, the currently used system by the Technological University of Dublin.
  + Analyse collected requirements to develop an initial system model.
  + Define the functional and non-functional requirements of the system
  + Outline the architecture of the system.
* **Design**
  + Identify a relevant design methodology to create the system.
  + Design the logical architecture.
  + Describe the initial physical infrastructure.
  + Use the relevant design methodology to implement the system design process.
* **Testing & Evaluation**
  + Describe testing and evaluation approach to each part of the system.
  + Provide the test plan in an organized and structured manner.
  + Identify evaluation methodology for each part of the system.
* **Prototype & Final Implementation**
  + Implementation of an **initial system prototype** (along with this report) for the **Interim** composed of the following:  
    - ***Web Application***
      * Implementation of a functional login system for users.
      * Access to the relevant modules for users.
      * Access to the relevant notes and assignments for users, with lecturers able to create / attach uploads to both.
    - ***Database***
      * Designed to support prototype implementation.
      * Inclusion of custom data to mimic student and lecturer data.
    - ***Local Network***
      * This prototype system will run on two separate virtual machines.
  + Implement a **final system** (along with the final report)to include the following:  
    - A web application supported by the database to have the following functionality:  
      * ***Students***
        + Login & Registration
        + Access to Modules
        + Access to Notes
        + Access to Assignments
        + Access to Grades
        + Access to other General Features (Accessibility & Tools etc.)
      * ***Lecturers***
        + Login (pre-approved system)
        + Access to Modules
        + Ability to Upload Notes
        + Ability to Upload Assignments
        + Ability to Grade
        + Limited ability to customize certain areas
        + Access to Other General Features (Accessibility & Tools etc.)
    - Implement a performance monitoring system using Prometheus & Grafana for administrators.
    - Limited access for Administrators to deal with issues in the Web Application, Database and Monitoring System.  
      * Web Application Administrator Panel
      * Database Access
      * Monitoring System Access
* **Issues & Future Work**
  + Determine if there should be additional features.
  + Determine if features should be improved or removed.
  + Identify key issues that have arisen during development of the prototype.
  + Determine if there are more potential issues down the road and how this project may have to adapt to them.

Once these milestones have been achieved, the project will be completed to a satisfactory and comprehensive degree.

## 1.4 Project Scope

The scope of this project solely targets Brightspace and the current registration system in place, it does not seek to replace the website or current Office365 facilities used by the university, instead it seeks to harness them into the design.

For example, the use of Microsoft Teams in place of the Bongo Virtual Classroom, would ensure that off-site learning can still take place, if the situation ever arises again, just as we found ourselves during the pandemic.

This would be an example of removing certain aspects of bloated design features, such as students finding themselves switching between Teams and Bongo, depending solely on lecturer preference, while trying to keep an eye on the calendar of both.

With no access to the university network security, nor the official data records of each student and lecturer, the creation of fabricated data to suit the needs of the system will be necessary.

## 1.5 Thesis Roadmap

The literature review will take you through the initial research phase of the project, including the alternative solutions found in other universities, the technologies needed to support this project, technologies that could be considered for future implementation, alongside similar applications and systems that have been developed by former students.

The system analysis will focus on the framework necessary to develop the project, such as the fundamental requirements needed for the user experience, for both student and lecturer, how this project could accommodate the influx of hundreds of users at once, and what would be required to abide by the necessary guidelines for accessibility, availability, security and data integrity.

The next phase, is the system design itself, focused on the details of how this system is to be developed, the chosen design methodology, alongside the physical and logical architecture design to support the wide array of features that will be implemented.

Following this, will be the outline of the Evaluation & Testing phase, which will be essential in determining the functionality of the system following implementation, alongside gauging what is needed to be added, and what can be removed, without compromising the project.

Next, will be the physical prototype of system itself, which has been created for the Interim. It will implement various aspects of the previous phases, especially regarding evaluation and testing.

As the final part of this report, before the references and appendices, there will be a review of the issues faced, problems that may arise down the road based on what has occurred, and how this project will need pivot to tackle these areas.

# 2. Literature Review

## 2.1 Introduction

In this chapter, a review of relevant research materials relating to this project will take place, including papers on alternative existing solutions for this system, such as the currently available Virtual Learning Environments, including Brightspace, which is in use in TUD [3].

There will be a review of the technologies considered for developing this system, such as HTMX and Django. There will also be a review into the necessary areas relating to the development of this system, regarding important development guidelines, such as the General Data Protection Regulation.

As an overview, a Virtual Learning Environment, alternatively known as a Learning Management System, are software services designed to provide online learning, hybrid learning between on-site and off-site learning, or generally enhance on-site learning for students at universities [4].

At the time of the formation of the Technological University of Dublin, the individual institutions used separate Virtual Learning Environments, with the Dublin Institute of Technology using Brightspace, while both the Institutions of Technology in Blanchardstown and Tallaght used Moodle. Ultimately, D2L Brightspace was chosen as the main Virtual Learning Environment to be used in the newly amalgamated university [5].

Universities in Ireland, contemporary examples to TUD, being the University College of Dublin, and the University of Limerick, are engaged in active research and engagement with supporting their Virtual Learning Environments on a yearly basis [6] [7].

The consistent implementation of a Virtual Learning Environment is clearly of vital importance in the day-to-day management and support of student studies from the evidence shown, with the support necessary during situations such as the COVID-19 Pandemic.

## 2.2 Alternative Existing Solutions

The following solutions to this system are contemporary Virtual Learning Environments, developed and used internationally [8], along with being found in the Republic of Ireland, in universities such as the University College of Dublin [9], Dublin City University [10] [11], Maynooth University [12] and Trinity College [13], along with the Technological University of Dublin [3].

While some of these solutions use more basic forms of the provided VLE platform, solutions such as Loop, which is used by DCU, are built off these platforms (Moodle in this case), with a far more unique experience, designed for the individual needs of the student, by combining multiple software services into one.

While an aim of this project is to condense the services required to provide the necessary tools for the student, it is important to note which direction each solution has taken to implement said tools.

**Brightspace**

Used by the Technological University of Dublin, the D2L Brightspace Virtual Learning Environment is focused on ‘learner achievement’ and bringing an international solution to universities [14]. It can provide access to individual modules, extensive access to previously studied modules, clear notifications regarding upcoming assignments, released grades, along with a large array of features that can be implemented [15] based on need, such as discussion forums [16], class progress [17] and a portfolio [18]. The D2L Brightspace Community provides a vast supply of knowledge on accessibility, issues faced, partnerships and training, with the level of features incorporated requiring a lengthy amount of time to understand [19].

The advantages of Brightspace include the complex tools available for learning, such as the Bongo Virtual Classroom, along with the accessibility tools that have been provided, including on-screen keyboard support and captions.

The disadvantages of Brightspace revolve around the vast layout of available features, some of which will never be used by an institution or user, and these features may take a while to find, then learn and use. This contradicts Nielsen’s Law of the Internet User Experience.

**Moodle**

Used currently by both Dublin City University in a extensively customized form (the latter mentioned Loop) [11] & by Maynooth University in a more basic form [12], alongside the previous institutions of the Blanchardstown / Tallaght Institutes of Technology, the Moodle Virtual Learning Environment, is a extensively customizable platform, to support the individual needs of a university. Like Brightspace, it provides access to individual modules, access to previously studied modules, clear notifications regarding upcoming assignments, released grades and more.

It, however, provides a more extensive platform for discussion forums between users than Brightspace, offering multiple subscription methods such as optional subscription (can choose whether or not to subscribe to discussion), forced subscription (forced participation in accessing discussion), auto subscription (initial added participation to the discussion but option to unsubscribe) and disable subscription. It also allows for discussions to be graded (this is included in Brightspace), alongside multiple types of forums of discussion (which Brightspace does not offer as extensively) [20].

While it does provide a large amount of features, it lacks the extensive officially supported features provided by Brightspace [21], instead opting for a wide array of plugins created by the community, which are not officially verified for use by Moodle [22].

The advantages of Moodle include the in-depth customization provided to modify the appearance and structure per implementation. This is displayed later in the creation of Loop, a custom solution for Dublin City University, which amalgamates with more than one piece of software services.

The disadvantage of Moodle is that a large number of potential features have been created by the community, and are not officially supported by the platform, but by the relevant community creator. While many could potentially be designed to a reliable standard, this is not guaranteed, if a plugin is needed and chosen for implementation.

**Blackboard**

The core of the Virtual Learning Environment used in Trinity College Dublin, it provides much of the same features previously mentioned, such as module access, previous module access, notifications for upcoming assignments and released grades.

Blackboard Ultra [13] is used by the lecturers to provide learning materials such as notes and videos to the students, communication with said students, the creation of online assignments, alongside provided integrated tools such as Turnitin (plagiarism detection software) [23], Panopto (video capture service) [24], and Class Collaborate [25].

Where Blackboard proves itself, is the number of available features that eclipses Brightspace, Canvas and Moodle, and the rate at which these features are being released [26].

It provides new and complex features such as an AI Design Assistant [27], Content Designer with AI Suggestions [28], and the Anthology Digital Assistant [29].

The main advantage of Blackboard is that it provides the most tools out of any available Virtual Learning Environment of equivalent size and provides these tools at a far faster rate per annum.

The disadvantage of Blackboard is that the pricing is greater than the other contemporary solutions available on the market.

**Canvas**

Used internationally, at home and abroad, in universities such as the University of Galway [30], Queen’s University Belfast [31], and Oxford University [8], the Canvas Virtual Learning Environment focuses on being as accessible as possible, being available in over a hundred countries, in thirty three different languages, with the infrastructural support for up to six million concurrent users.

Like each previously mentioned Virtual Learning Environment, it provides access to modules, learning materials, assignments and grades. However, it provides a parental option, compared to just the typical student and lecturer, which would be useful for younger tiers of education [32].

It provides tools such as SpeedGrader [32], which has extensive grading tools, including comments and annotations, Gradebook [32], which offers a clean view of learner progress and the ability to adjust weightings, and the organization of data. Canvas Data provides access to event-level data files to assist in the construction of custom dashboards, advanced queries, designed for potential integration with a complex data warehouse [32].

With a comprehensive global approach to accommodate most nations and their languages, it is clear why Canvas is used so extensively across the globe.

This is the advantage of Canvas, the sheer level of accessibility to the platform, with a clear catering to supporting as many customers as possible.

However, the disadvantage of Canvas is that it does not provide the same level of features as other platforms

**Loop**

A custom solution created for Dublin City University [10] based on Moodle, this platform serves as an amalgamation of learning tools, including a mobile application available on Google Play and the Apple App Store, featuring offline browsing of course content.

While also providing access to modules, previous year content from those same modules, access to assignments and grading, with Loop Class (the core part based on Moodle) [33], Loop Connect (powered by Zoom [34]) is provided, as a virtual classroom, for communication online between student and lecturer when necessary, with Loop Check (powered by Ouriginal [35]) provided to detect and challenge plagiarism in student work, and finally Loop Reflect (powered by Mahara [36]) which is a provided ePortfolio system, to support those courses that need it.

Compared to the other solutions, Loop was clearly designed for the specific needs of DCU, incorporating multiple software services (not just Moodle), yet condensed to avoid the bloat of features that is prevalent in the other services.

The advantage of Loop is that the design is catered specifically for the university, with the institution choosing to take a number of typical software tools, combining them into one platform. This has allowed them total freedom on how it is implemented,

The disadvantage of Loop is that the maintenance, like this project, must be provided for in a greater manner than other solutions, due to the internal development of the project.

Out of all the solutions, Loop is closest in nature to this project, but still uses available software solutions as a foundation, bringing different disadvantages to the table.

**Alternative Existing Solutions Conclusion**

While technology has expanded the scope of these tools, it has led to each solution providing nearly identical tools, with the main difference between the Virtual Learning Environments being the user interface, with how these tools are being accessed.

Blackboard provides the greatest amount of features, for example, but many of these will not be necessary for the engagement of studies for most students, merely designed as assistant tools to help the needs of certain lecturers or students, and they are not necessarily tools of accessibility, or a clear enhancement to the user experience.

This growth has led to bloat, the complexity of implementation, and also an increase in service price. With the struggle to implement a registration system in the summer, that upended the use of Brightspace for weeks, there is a clear avenue for a potential in-house solution.

In a study comparing a number of the listed Virtual Learning Environments [4], including Canvas, Brightspace, Moodle and Blackboard, the difference between them is shown to be minimal, with all aiming to provide the same fundamental features – access to educational content.

The growth of Canvas globally has stood out, with their focus on accessibility, through the wide array of nations and languages that it is available for. While Moodle was once acknowledged as ‘the most powerful platform in terms of its tools and their capabilities’, there has been a clear converging of features between all available Virtual Learning Environments. While one platform may perform marginally better with one feature, another platform performs marginally better with another.

The pricing of these solutions is another matter. The subscription pricing (based on estimated cost per user) differs, with Brightspace typically ranging from £8 - £15 per user [37], £13-20 per user for Blackboard [38], £8.73 per user for Canvas [39], with Moodle being open-source with the option for hosting (no clear pricing except for a low amount of users) [40]. These subscription prices are based on the roughly 28,000 students currently studying at TUD [41], and these prices do not account for initial implementation or potential maintenance down the road.

The conclusion from this, is that each platform provides largely the same outcome of tools and features and are very expensive.

This project goes in the direction of developing and hosting a solution directly, and depending on the cost for development (if developers were to be hired by the university to engage in such a project, and the potential maintenance staff required for hiring), this could turn out to be cheaper implementation, even if it was necessary to host the solution off-site and on the cloud. There will be an investigation into pricing later in the review, when services such as Amazon AWS, Microsoft Azure and Google Cloud are covered.

Eagna will aim to provide the necessary tools and features required to engage in studies across all courses, but it will focus on the needs of accessibility, along with providing a solution to the current registration issue, along with condensing everything into a platform that will use the additional services available, such as Microsoft365, in an attempt to limit cost but providing a viable solution at the same time.

These mentioned platforms will grow, in available features, availability and pricing (with both Brightspace and Canvas stating a clear increasing cost per annum in their documents), but an independent solution can be managed based on the swelling and lessening needs of the individual institution. Microsoft365, formerly known as Office 365, has been available since 2010 and has evolved since its inception.

Developed and maintained by one of the largest technological conglomerates, it provides a stable support system for the university email system, the concept of this project, along with the tools it offers to help match the available features of the other Virtual Learning Environments, such as using Microsoft Teams as a built-in virtual classroom.

## 2.3 Core Technologies

There are a number of technology types that will need to be combined in order to create a system as complex as this. With a web application as the primary core of the system, there will be a need for an array of front-end and back-end technologies to support the nature of the application, to enable the needs of the users. To support this web application, there is a requirement for a database, to store the relevant records and data that can be accessed by the web application, in a secure manner that supports modularity and supports a separation of concerns, in relation to the separation of nodes in the distributed system. On top of this, there will be a need for monitoring technologies, to help manage the reliability, availability and integrity of the system. There will be a review of virtual machine technologies, to support the initial prototype, a review of container orchestration technologies, should there be a future deployment, for an expansion on reliability and availability, and finally a review of cloud services, should there be a need for deployment off-site, in place of on-site infrastructure.

**Web Application Technologies**

***Front-End Technologies***

The technologies of the front-end focus on the structure, presentation and behaviour of the user experience. Using these technologies to form the basis for the User Interface, they will interact with the back-end technologies to form the visual core of the web application.

With markup, styling and scripts, the choice of technologies will have a fundamental impact on how you design and deploy a web application.

**HyperText Markup Language (HTML)**

This technology forms the foundation of the World Wide Web as we know it, with the development of HTML regulated by the World Wide Web Consortium (W3C) [42]. It is a markup language, meaning it uses tags to define the structure of a document, and how the various parts of the document work together. With HTML, a browser, such as Google Chrome or Mozilla Firefox, can understand and parse the content to be displayed to the user.

As of 2017, HTML5.2 is the latest version used [43], and while it forms the structural backbone of the vast majority of the web, the use of technologies such as CSS & JS have become fundamental design instruments to edit both the presentation and behaviour of HTML webpages.

This technology is necessary and fundamental to this project. It is unique in the fact (alongside CSS) that the advantages and disadvantages of HTML for this project are irrelevant, as there is no sustainable alternative for what it can provide [44].

**Cascading Style Sheets (CSS)**

Used on top of HTML, Cascading Style Sheets are used to style the webpages you create, forming the backbone of the presentation of your website. Also standardized by the W3C [45], it enables the design of responsive webpages, supported by the variables, logical properties and layers to provide scalability while remaining relatively easy to maintain.

It remains the global foundation for the presentation of the user experience on the web, natively supported by browsers.

Like HTML, the advantages and disadvantages of CSS are irrelevant, as it also forms the standardized backbone of the web for presentation, being a necessary part of most web applications [46].

**JavaScript (JS)**

JavaScript is one of the most popular programming languages used on the web, allowing pages to respond dynamically to the user. The reason it is used globally is due to the compatible nature of the language, being supported by most browsers and devices, with a vast array of frameworks, libraries and tools to help support development [47]. It offers functionality to your website like few other frameworks can [48].

The advantages, in relation to this project, of JavaScript, include the vast compatibility it offers across browsers and devices, alongside the huge ecosystem that supports it.

The disadvantages, in relation to this project, of JavaScript include the underlying difficulties of writing with it, potentially slowing down development and introducing bugs that may go unnoticed without extensive testing.

**HyperText Made Xtreme (HTMX)**

HTMX is fundamentally a lightweight JavaScript library, to use in place of vanilla JavaScript, that allows the developer to add modern and dynamic behaviour to HTML with a focus on minimizing the code necessary [49].

The main usage of HTMX is for fast iteration and accessible pages without needing a complex front-end framework. With a focus on stepping away from the design process of a single-page application (SPA), where a web application would interact with the user to dynamically rewrite the current web page with new data from the server / database, it instead takes the approach of annotating links and forums with attributes that handle partial page requests, swaps and transitions through Asynchronous JavaScript & XML (AJAX), WebSockets or server-sent events. This means that HTMX is naturally light on the client-side state, focusing instead on keeping the authoritative state within the server.

The advantages, in relation to this project, of HTMX, are that it pairs naturally with back-end technologies such as Django and Ruby on Rails, as a major part of the design philosophy was to remain framework agnostic. It also focuses on development for web applications with modern & dynamic behaviour at faster speeds. The central focus on the server-side state, as opposed to the client-side state, supports the nature of this project, which is focused more so on the management and access of data, as opposed to a complex user experience

The disadvantages of HTMX, in relation to this project, primarily revolve around a smaller ecosystem, in relation to both supported frameworks and available examples of usage.

**React**

The more mainstream approach, compared to HTMX, React is a JavaScript library for designing interactive user interfaces, organized into components and a clear data flow. Unlike HTMX, React was designed to focus on the client-side state, the core design philosophy in providing a rich interactive user experience (including the design philosophy and process of the previously mentioned single-page application), with less weight on the server-side state [50]. It does, however, offer the option to work with both.

The advantages of React, in relation to this project, include the sheer amount of functionality and features that it can provide [51], along with extensive documentation for the use of it [52]. The client-side centric state supports a perceived increased speed in user interaction (depending on the user’s machine), even with poor internet connection, also leading to a lower traffic load on the server.

The disadvantages of React, in relation to his project, include a greater exposure to security and data risk, with more data being stored in the browser due to the increased client-side nature, increasing the risk of data being stolen or misused, and an increased complexity on providing thorough management of the segregation and management of data. With the amount of data that will be potentially processed in this system, it poses a problem for the user experience, as React suffers when dealing with large data sets at once [53]. It is also very complex in regard to implementation, due to the extensive nature of the available features and the underlying system to support them.

***Back-End Technologies***

The technologies of the back-end focus on the support, reliability and functionality of the user experience. Using these technologies to interface between the front-end and the database, they will interact with the back-end technologies to form the technical core of the web application.

With caching, query management, and authentication, the choice of technologies will have a fundamental impact on how this web application can function.

**Django**

This technology is a Python-based web framework designed for secure and maintainable web applications, with a focus on fast development [54]. It uses typical conventions, such as URL Routing [55], and comes with built-in features such as user authentication [56]. It was used in the development of massive web applications such as Instagram & YouTube [57].

The advantages of Django, in relation to this project, include quick development and deployment, secure authentication features and a massive support system, with a focus on code reusability and reliable maintenance [57].

The disadvantages of Django, in relation to this project, include a lack of support for unstructured queries, and the dependency on its own ecosystem of available resources and plugins in regards to development flexibility [57].

**Flask**

Like Django, Flask is a Python-based web framework, with a difference in focus on being lightweight, aiming for the title of being a ‘microframework’ [58]. It is the perfect framework for building prototypes and light applications, yet it still provides a sizeable ecosystem to support development with it.

Some of the advantages of Flask, in relation to this project, include the lightweight and flexible nature of the framework, primarily designed for smaller applications [59]. It is easy to learn and can be setup quickly.

Some of the disadvantages of Flask, in relation to this project, are that it does not provide similar built-in functionality like the authentication system of Django [59], requiring far greater independent design considerations, resulting in inconsistent implementations across applications designed with the framework, which can be an issue in trying to learn from other systems. By nature, it also requires a greater level of work in regard to maintenance, due to a lesser level of built-in features, in comparison to Django or the latter Ruby on Rails.

**Ruby on Rails**

This framework, unlike the other two, is more of a full-stack web application framework, built with the Ruby programming language. It is also used for quick web development, focusing on the ‘convention over configuration’ principle [57]. It is made for ease-of-use for the developer and supports the same level of integration tools as Django, such as for routing [60] or ORM [61].

Some of the advantages, in regards to this project, of Ruby on Rails, includes the built-in tools, as previously mentioned, a clear focus on productivity through based on developer ease of use and secure defaults (the practice of having the most secure configuration out of the box) [62].

Some of the disadvantages of Ruby on Rails, in relation to this project, includes the fact that performance can be slow, due to the focus on ease of development, which would have a knock-on effect towards scalability, and a lack of flexibility in regard to design, with technical aspects such as configuring routing taking longer to flesh out, in comparison to Django [63].

***Database Technologies***

The technologies of the database focus on the storing of data and records, essential to how the user will access the application. While the back-end will interact with the database to pull and store information, based on the interaction between the front and back-end, the database needs to be designed in order to support the quick, reliable delivery and storage of data.

**SQL**

Structured Query Language, otherwise known as SQL, is used by relational databases, with contain data in tables with rows and columns, linked together by keys (primary and secondary), with a predefined ‘schema’ (or design). It focuses on consistency and adhering to the ACID (Availability, Consistency, Isolation & Durability) principle, with a clear focus on structure [64].

*PostgreSQL*

A popular open-source object-relational database management system, it is well known for its reliability and standards compliance [65]. With over three decades of development, it is used in many different forms of software deployments, compatible with all major operating systems, and has a number of powerful features available [66].

The advantages, in relation to this project, of PostgreSQL, are powerful queries (great ability to pull data by design), a focus on data integrity and compatibility with the aforementioned technologies for web applications, such as Django. Given that it has been around for so long (since the 1990s), it provides a stable backbone for data storage backed by longevity, with a proven design that supports concurrency to help boost performance [67].

The main disadvantage, in relation to this project, will be the required complexity in schema design, in order to support the nature of the application.

*MySQL*

A widely used open-source relational database management system, it is maintained by Oracle and focuses on simple maintenance and productivity. Created also in the 1990s, it provides a stable and reliable solution for an ACID-based design [68].

The advantages, in potentially using this for Eagna, are that it has a deep ecosystem [68], has decent speed despite weak concurrency integration [67], and longevity with a wide range of documentation available [69].

The disadvantages, in relation to this project, are that it is a slower solution in comparison to PostgreSQL [67], while also requiring a similar level of deep complexity in schema design like PostgreSQL.

**NoSQL**

In comparison to SQL, NoSQL, or No Structured Query Language, focuses on flexibility and scalability. It does not adhere to ACID, instead it seeks to provide support for rapid change in data. Using key-value stores and document databases, the lack of rigidity provides greater support for inconsistencies in the structure of data [64].

*MongoDB*

A database technology utilizing documents, it stores flexible data in JSON-like records, in contrast to the tables of an SQL-based database technology. Released in 2009, it has gained recognition as a leading NoSQL solution, based on an intuitive data model that mimics an application object [70].

The advantages of MongoDB, in relation to this project, include an easier storing of data (a commonality among NoSQL solutions), a choice in combining structured and unstructured data, enabling design flexibility, alongside a vast database of documentation available [64].

The disadvantages of MongoDB, in relation to this project, stem from the nature of NoSQL. While it may be easier to throw everything in together, it lacks the clear mandatory structure that this application will need in order to access structured and maintained data [64], with more maintenance also required to keep in accordance with GDPR.

*Apache Cassandra*

Focused on resilience, through strong fault tolerance and linear scaling, and high write throughput (the ability to process large amounts of data at the same time), Apache Cassandra is an open-source distributed database used by thousands of companies globally, designed to support the storage of data across servers and regions [71].

The advantages of this technology, in relation to Eagna, include the focus on scalability and fault-tolerance, while also remaining flexible in regard to database design [72].

The disadvantages of this technology, in relation to Eagna, range from the data modelling issues stemming from the nature of NoSQL [73], alongside the encouragement for the use of unofficial tools such as the Kashliev Data Modeler [73] to help in streamlining schema design [72].

**Monitoring Technologies**

The technologies relating to performance monitoring will be solely for the maintenance of the application, to be reviewed by specialists concerned with the upkeep of the system. The typical users, the student and the lecturer, will never see this side of the system, but they will depend on it all the same to the use the application. Monitoring the connection between the application, database and outside world, it is essential in the modern day to maintaining availability and up-keep, to know when something has gone wrong, where it has gone wrong, being the first step in finding a solution.

***Prometheus***

A leading solution in the industry, Prometheus is an open-source monitoring system, designed to track the metrics of applications and infrastructure, ranging from the uptime of your servers to the average read and write throughput of your database, on whatever time interval is desired [74]. It collects data (like a database) with an exporter (such as node-exporter), which can then be used to determine the performance and reliability of your system.

The advantages of Prometheus, in relation to this project, range from the fact that it is open source, with access to powerful querying tools to pull metrics (PromQL), to a vast ecosystem of tools to help monitor your platform, including the latter mentioned Grafana, which is supported by the platform [75], alongside previous usage during a Site Reliability Engineering Internship at Mastercard, showing a clear example of industry usage.

The disadvantages of Prometheus, in relation to this system, includes the fact that the storage and memory requirements are sizeable (on top of what is necessary for a system such as this), while it provides only the bare minimum visualization tools, requiring other related software solutions to provide more human-friendly options of visualizing data [75].

***Grafana***

A visualization and dashboard platform, it is designed to take in metric data, transforming it into visualization to assist in maintenance and the upkeep of reliability. Also created as an open-source solution, it provides a single place to access and create any dashboard an engineer may need to display and review data relating to their system [76] [77].

The advantages of Grafana include, in relation to this project, the easy ability to visualize metric data, create alerts when certain parameters are met (such as dangerous / high load on a node in your system), the sharing / importation of dashboard setups [75], alongside previous usage during a Site Reliability Engineering Internship at Mastercard, showing a clear example of industry usage.

The disadvantages of Grafana, in relation to this project, are slim, with the only concern being the learning curve.

***Zabbix***

Created in the early 2000s, this monitoring platform provides the means to store, visualize and analyse metric data for your services and network, including built-in tools, unlike Prometheus [78].

The advantages of Zabbix, in relation to this project, include the fact it supports the aforementioned Grafana as well, and the storage of data within an attached relational database and provided rich visualization (as opposed to using Prometheus alone) [79].

The disadvantages of Zabbix, in relation to this project, include the fact that the ecosystem is far weaker than the Prometheus / Grafana configuration, and no previous experience in using it.

***Datadog***

A commercial monitoring solution, hosted on the cloud, Datadog, like Zabbix and Prometheus, serves as a platform to monitor the metrics and logs of your system. Created in 2010, it is a lightweight and easily integrated solution to help manage your system [80].

The advantages of Datadog, in relation to Eagna, include the fact that it provides a wide array of features for metrics, tracing, logs and real user monitoring (an insight into the performance of a web application for real users) [81].

The disadvantages of Datadog, in relation to Eagna, include the fact that the free version is limited in scope (a one-day data retention period) while also lacking the wide array of aforementioned features in this free version [82].

**Virtualization Technologies**

Forming the backbone of this prototype, virtual machines can be used to deploy isolated operating systems, either to access another operating system on your machine, or to mimic deployment on separate hardware devices. They are often used for prototyping concepts like distributed systems (with each machine acting as a node) and networks (linking each machine to one another), where their contents can then be later deployed after proper testing to accessible infrastructure, either on physical hardware locally, or accessible through the web by way of cloud services. There will be a review of the two main solutions that have been used in this course [83].

***Oracle VirtualBox***

Maintained by Oracle, VirtualBox is a dynamic virtualization software compatible with both Intel & AMD processors [84].

The main advantage of Oracle VirtualBox for this project, is that it is open-source, while providing a powerful array of features, such as managing the network between these machines, an easy setup (including the deployment of operating system images like Ubuntu Server) with an efficient use of memory resources [83], alongside efficient performance for small to medium sized projects [85]. It also has a large community and a vast amount of documentation / training to help in using it [84].

The disadvantages of Oracle VirtualBox, in relation to this project, are that it provides an older GUI and can be quite heavy on storage resources, while also being less popular in professional environments, and slightly slower, in comparison to VMWare, when working with larger block-sizes (essentially larger amounts of data).

***VMWare***

VMWare, supported by Broadcom, is a commercial virtualization software platform that also provides cloud infrastructure and security considerations [86]. It has a couple of options in terms of software: Workstation, similar in nature to VirtualBox, allows for the deployment of operating systems in an isolated environment on your machine, Fusion, which was designed for users of MAC OS [83].

The advantages of VMWare in relation to this project, include the fact that it is a more complete solution in terms of the integrated features available and the customization provided, including the previously listed features of VirtualBox, while also providing better performance for larger block sizes [85].

The main disadvantage of VMWare is that it is not open source, requiring all maintenance and technical solutions be provided by developers at Broadcom in a timely fashion, while it was also not as commonly used by lecturers in TU856, in comparison to VirtualBox.

**Core Technologies Conclusion**

In relation to the front-end, the use of HTML, CSS and JavaScript functionality through HTMX, will form the basis for the front-end of this web application. The choice to use HTMX, in place of React, is due to the nature of the complexity of React, its primary support for the client-side, and the fact that it is far heavier in code in comparison to deploying similar functionality using HTMX, with a significant increase to performance [87].

HTMX focuses on keeping state and logic within the server, whereas React is designed to manage state and logic primarily with the client. Where you would choose React for complex features and the sharing of client logic, HTMX is designed to render from the server (supported by a back-end technology such as Django or Rails), with a focus on speed of implementation, simplicity in design and progressive delivering of results during development, which will be key for designing this project for the final year.

The project does not revolve around complex user interaction, it is meant to be simple: login or register, access your modules, read or download the material there, upload your assignments, and access tools provided by other services such as Microsoft365. The ‘complex’ user interaction will only consist of lecturers updating text, and other minor details.

In relation to the back end, the use of Django supports the chosen technologies of the front-end, and there is plenty of documentation available with usage alongside HTMX [88]. This setup also supports the chosen technology of the database, PostgreSQL.

Django is well suited for productivity, scalability, performance, with a large ecosystem that can support web applications, making it the perfect choice for this project [63].

The choice to use SQL, first, over NoSQL, as the structure and definition of SQL will better support the nature of the data relevant to this program. The structure and integrity of this data is essential, in relation to both GDPR and also how user records will be implemented.

In regard to PostgreSQL, over MySQL, it is clear from this reviewed study [67] that PostgreSQL’s support for larger data sets, with an excellent integration of concurrency, gives it a clear advantage in performance in most regards, which is the determining factor for this project between two solutions for an SQL database technology.

In regard to Monitoring Technologies, the use of Prometheus and Grafana provides a thorough and modern solution for managing availability and maintenance. Having used it during an internship at Mastercard with the Site Reliability Engineering team, it is a proven industry setup.

In regard to the use of Virtualization Technologies, VirtualBox has been chosen due to its use in the course, the performance it can still provide for small and medium block sizes (which works for prototyping), and that it is relatively easy to use and setup.

With the chosen technologies, Eagna can be designed efficiently, rapidly and with a clear structure for review.

## 2.4 Requirements

In this section, there will be a review of the relevant areas of functional / non-functional requirements of the system, to gauge the areas of importance relating to the development of a system such as this.

**Availability**

When your software is not accessible to the user, due to downtime or errors, then it cannot be used, meaning the management of the Availability of your system is essential. There are three key areas in relation to availability that will be reviewed:

* **Redundancy**  
  This is, in essence, the provision of multiple copies of the same software / functionality block in a system, whereby if one deployed version fails, another exact replica can immediately be deployed in order to fill the gap created by the now unavailable node. In this case, it would be two copies of the web application, deployed on separate servers / virtual machines, when one fails, the copy is immediately deployed. While this will not be physically implemented with the prototype, it is supported naturally by the design of the distributed system, due to the isolated environment of the virtual machine. What is contained within that virtual machine would be deployable as copies on multiple servers, ready to be deployed when failure is noticed / occurs, at a moment’s notice, supported by the use of a container orchestration technology such as Kubernetes [89].
* **Observability**  
  This is, in essence, the ease of understanding the health of a system, the ability to gain an inferable insight (based on the viewed end result) on faults and errors that have caused the failure in the system, provided by the nature of the system design and tools implemented in the system. In this case, it would be the setup of the monitoring system, which would allow an engineer who is responsible for maintaining the system, to actively monitor and understand why an error has occurred, and where it has occurred. General areas of note for Observability would include the review of metric data for latency between system nodes (in the case of the prototype, the speed of the passage of data between each virtual machine, and the overall understanding of the health of each system node (such as detectable errors or downtime) [90].
* **Recovery**  
  This is, in essence, the designed ability and structured outline in how the system should be recovered in the event of a fault, to help re-deploy the failed system part as fast and efficiently as possible. In this case, the failure of a system part, such as the web application server / virtual machine, would have an outlined path and means to be fixed, before re-deployment. While the aforementioned Redundancy will help with preventing the downtime of the system, there is cause for concern that the same error may occur on an exact copy of that system part. This will be the hardest to take into consideration and is usually a fault caused by mismanagement of design and implementation [91].

**Performance**

If your software does not run well, the user will have a difficult and frustrating time trying to do what they need, in a fashionable manner. There are three key areas that will be reviewed in the following section:

* **Caching**  
  Previously mentioned in the ‘Potential Technologies Researched’ section, Caching is used to store copies of frequently accessed data, temporarily, in a high-speed location, in order to limit the strain on the database. PostgreSQL natively provides the ability to cache data, through the caching of data in tables, indexes (for searching through data records), and the execution plans of queries. While it helps in improving the performance of a system, caching introduces the potential for out-of-date data to be pulled, should data from the cache, that has since been updated in the actual data storage, be used instead of retrieving the updated data [92]. There is also built-in functionality for caching with Django to support the web application [93].
* **Concurrency**  
  Also being briefly mentioned but not expanded on when reviewing PostgreSQL, this will be essential in this system. Concurrency is the ability of software to process more work in the same moment, thereby increasing the overall performance of the system. If only one user at a time could do things using your system, then the dozens, hundreds, or thousands behind them would have to wait. A system being used by multiple users at once must be able to manage the individual actions of each, at a respectable speed, in a given moment. The ability of used software technologies to apply concurrency with simultaneous execution of tasks (like in PostgreSQL), is essential [67].
* **Query Optimization**  
  Using indexing, join orders and access paths, Query Optimization focuses on traversing the least amount of data possible, in order to speed up the retrieval process of data. The use of queries such as ‘SELECT \*’, where-by selecting as much data as possible when performing query, requires a larger set of data. Using indexing, for example, allows the fastest retrieval of data by jumping directly to a matching row, as opposed to scanning an entire table of data. There a number of types of indexing available, such as B-trees and Generalised Search Trees (GiST) [94].

**Security**

To ensure that the user’s data is used against their will, the security of any application is vital to the operation of your software, and for the day-to-day lives of your user. There will be three key areas relating to this project for review, which will now be listed:

* **Authentication**  
  This is the verification that the user is who they are claiming to be, while they try to gain access. In the case of a student, when they are entering their email and password, this will need to be verified as accurate, based on the current data stored in the system. Django includes a built-in authentication system, which will form the core part of this non-functional requirement of the system [57]. How Django will take in user input for email and password, how this is verified by Django, with the relevant data verified by the PostgreSQL database, is essential for this project.
* **Encryption**  
  In order to secure storage and transmission of data, so that it cannot be intercepted or stolen by bad faith actors, encryption will be essential. PostgreSQL can use SHA-256 Encryption for the storage of data such as the passwords of users. While Django provides out-of-the box authentication, it does not provide encryption, which may have to be tackled with the usage of TLS [95] in a future deployment.
* **Access Control**  
  In concurrence with Authentication, this is a part of the system design whereby access is restricted based on the user. Authentication may verify that the user is who they say they are, but this prevents users logging in and having access to areas of the system they should not have access to. An example of this would be a student logging in, accessing their modules, and only seeing their relevant assignment submissions, they should not have access to their fellow student’s assignment submissions, nor the details of their grades. This will primarily be accommodated for based on Django’s User Authorization, with database querying linking areas based on user login credentials [96].

**GDPR**

The General Data Protection Regulation is set of strict rules, outlined by the European Union, that went into effect on May 25th in 2018 [97]. It was designed to give European citizens control of their data footprint, and it applies to any form of personal data being collected on the individual.

The focal point of GDPR are the transparency principles, where Article 5 determines that data must be limited based on purpose, that only so much should be stored, that this data remains confidential, and that the integrity of the data is not compromised (destroyed or modified) [98].

This can apply to any of the following potential points of data in this system: usernames, email addresses, passwords and grades.

If there were to be a breach in any of these outlined policies, which would be a risk to the individual user, there would have to be an incident report created immediately (or as soon as possible), with notification to the Data Protection Commission required with 72 hours [99].

There are other areas to be mindful of, including the storage of data outside the European Union. An example would be the EU-US Data Privacy Framework [100], which outlines the legal requirements in relation to the storage of an EU citizen’s personal data in the United States of America, which could be important, should the system be deployed on a US-based cloud server, through the likes of Amazon AWS or Microsoft Azure (if the chosen location for the server is inside the United States).

If this system were to be launched, with real student and lecturer data, then Articles 35 & 37 would apply, where-by a Data Protection Impact Assessment would have to be engaged in before the launch of the system, to identify and minimize the potential risks to user data [101], alongside the appointment of a Data Protection Officer, to verify this process [102].

In conclusion, the avoidance of storing personal data in unencrypted logs, loss of data due to backup failures, and the lack of safeguards in regard to the deployment of data are essential, if this were to be deployed in real-time. While this project will not be using the real data of students, it will be designed in acknowledgement of these requirements.

**Accessibility**

The most important metric in relation to the user’s experience, Accessibility determines the ease of use of the web application, for your general user, and for those with learning difficulties or physical disabilities. As there are many possible learning difficulties and disabilities to consider, it is best to split them into a number of categories, with information gathered by the World Health Organization (WHO), Enable Ireland, and other sources, to try and provide as much reach in this research as possible:

* ***Visual Impairments***  
  Playing a critical role in our every-day lives, our vision can be taken for granted, it is also the most important part of using our devices and therefore accessing this project.   
  According to the World Health Organization, at least 2.2 billion people suffer from some form of a visual impairment [103]. Conditions can include near-sightedness, far-sightedness, cataracts, glaucoma, colour blindness and more.
* ***Hearing Impairments***  
  Another vital part of our collective senses, we consider our hearing very important to our everyday lives, for communication, enjoying music, and so much more. To listen during your lecture, or to watch an online video uploaded by lecturer, it is essential.  
    
  According to the World Health Organization, roughly 430 million people require rehabilitation for their disabling hearing loss, with this number expected to grow to 750 million by 2050 [104]. Generally, the loss of hearing is same end result, with the use of hearing aids, cochlear implants and more being used to help tackle the issue.
* ***Mobility Impairments***  
  With a wide range of different mobility impairments, which may impact walking or moving your arms, this is a difficult disability to account for in a web application. A core aim of TUD is to provide the support necessary to avoid an educational disadvantage for those with disabilities such as these [105], so there will be an attempt to account for these disabilities where possible.
* ***Speech Impairments***  
  There are a number of speech impairments, including articulation disorders, where a student may struggle to speak in the way they intend, phonological disorders, where there are patterns of being unable to produce sounds the way in which they intend to, and motor speech disorders, where the student struggles to plan and co-ordinate their intended speech patterns [106]. While this is typically found in younger people, such as pre-university ages, this is not always limited to that scope. The features of this system will not impact those with speech impairments, except for the usage of extra tools that will be accounted for, such as Microsoft Teams, therefore it’s inclusion should be noted.
* ***Learning Disabilities***  
  A learning disability disadvantages a person in their ability to learn and develop as other humans do. With causes ranging from genetics, brain injuries and more, there are many different types [107]. Examples include Dyslexia, Dyspraxia and more.   
  There will be students in TUD who are affected by difficulties such as Dyslexia, so it will be important to try and accommodate them where necessary.
* ***Neurodivergences***  
  Considered a vague umbrella term for a number of potential different disabilities or difficulties, it is important to include for personal identification, especially for the basis of a future questionnaire, for those may not strictly be clearly disadvantaged, but may approach tasks or education differently, or in a non-neurotypical way. For those on spectrums, such as with Autism, for students with Tourette’s, or even those with mild to limited ADHD or Dyslexia, where it may not extremely hamper their learning experience, it is important to accommodate them, as they may feel still at a disadvantage to their peers in some areas of learning [108].
* ***Long Term Illnesses***  
  These are known as noncommunicable diseases, and the WHO has determined that roughly 43 million are killed each year by them [109]. Affecting all age groups, not just the elderly, they make life very difficult for those that bear them. In relation to this system, the overall ease of use, for those who wish to engage in studies regardless of their condition, for academic achievement, is important. Examples include respiratory diseases, cardiovascular diseases and cancer.
* ***Mental Health Conditions***  
  With nearly 1 in 7 people, according to the World Health Organization [110], living with a mental health condition, it is important to consider the students who may struggle with their day-to-day lives, and how it may impact their studies. Examples include Anxiety disorders, Depression, other disorders such as Bi-polar or Post Traumatic Stress Disorder.

In a survey conducted by the Disability Federation of Ireland in 2022, including details from the 2022 Census, there were reportedly 1,109,557 people with a long-lasting condition or difficulty, while 643,131 people reported they had a disability [111]. This shows a clear need to accommodate those people in this project, as there are probably numerous students who would identify with this report.

**Requirements Conclusion**

There are a vast number of potential requirements for this system, and there will have to focus on the implementation of those feasible, and based on the prototype, as a single student.

This research was still important, as it shows the importance of many key areas that are needed to be analysed for a learning platform. If the experience of the user is impacted by a design fault, failing to accommodate for their needs, it is on the shoulders of the developer to correct this.

## 2.5 Existing Final Year Projects

These are the final year projects that have been reviewed, that will provide a basis for setting the standard of the project report, and how they may have developed similar solutions in regard to technology usage.

**Suaimhneas - A responsive web application for anxiety management and wellbeing**

Developed by Ben McCormack, this web application was designed to help people manage anxiety using CBT techniques, recording their daily moods, with personalized feedback and a recommendation system.

His research is vast, diving deep into both the technologies he should use, and the experience for the user, and given that it is similar in setup to Eagna, in relation to the technologies and system he built, what he has done here could be a great help to understanding the level required.

While it is not relevant to this project idea, his ideas around creating the registration system, and the algorithm he may use, is quite complex, how he outlined the use of the cosine similarity algorithm as his final choice was interesting.

He used PostgreSQL as his database, Django as his framework for Web Development, JavaScript as his Front-End Technology, alongside using Docker, HTML & CSS. He also selected Digital Ocean as his Web & Database hosting service. (Figure 2.3)

Similar technologies have been reviewed, including Django, which was something that was ultimately chosen for this system. Overall, this project seems to be on the stronger side, in most regards.

His research was complex, thorough, and the basis for his design / implementation on this research was equally as impressive.

The main weakness would be the user interface, which could be improved upon a decent amount, making it softer in appearance and more accessible, the Mood Check-in Wireframe sticking out, along with the accessibility / readability of text across pages such as Create / Edit Goal Wireframe and View Individual Client (all found in the Appendices). [112]

**An eLearning system for Anonymous Feedback Sessions**

The second project reviewed was developed by Daniel Hogan.

An older project, this was designed as an eLearning system that would provide the ability to give Anonymous Feedback. His dive into Nielsen’s Heuristics was deep, using it to examine the main operating functions of his project, such as the Lecture Tools, Poll Everywhere and E-instruction’s Student Response System.

He also provided an interesting overview of the technologies he was researching, such as a comparison between SQL and MongoDB commands / queries. His architecture research was also impressive, especially his overview of API routes. He used a central node server, storing all the system data and providing API & Web Socket events to manipulate his data, using MongoDB and Mongoose as his ODM. He also used Bootstrap as his main web development technology, linking it to the server functionality.

There are more weaknesses in this project than the last. Despite the idea being more similar to this project, in comparison to the first project, that project has a far more expansive implementation.

This project done well with certain areas of the research, such as Nielsen’s Heuristics, but it appears to have suffered from a lack of research into the full implementation, and what would be necessary for important areas such as Security and Availability – something which will have to be avoided.

Given that it is older, leeway would have to be given on what could be expected, as some of the issues he ran into, such as saving session status, could be more achievable with what technologies are available now, and in comparison, to the first project reviewed.

[113]

**Existing Final Year Projects Conclusion**

The main points gathered from these projects is the depth of research found within the reports.

Both are clear, detailed, and well-structured, delivering a great overview of their relevant project design and implementations, with the research to back them up.

They have set a clear standard in quality in their reports, providing inspiration for the final report of Eagna, alongside the level of implementation that should be aimed for.

The first project, when initially reviewed, gave the inspiration into reviewing Django and deciding to implement it into this project design, due to relevant features such as the dynamic creation of a database schema, based on the design implemented in Django. It uses very similar technologies, in comparison to this project, and will be helpful during the project process.

The second project, while straying away from the Eagna technology stack (such as using a MongoDB / NoSQL database implementation), provides an insight into an eLearning system with a general similarity. The main points of this project that is relevant to Eagna are the requirements gathering, in particular, the non-functional requirements such as Nielsen’s Heuristics and the research into Security & Availability.

## 2.6 Overall Conclusion

It is clear, based on the evidence, that there are plenty of solutions and avenues to take in developing a Virtual Learning Environment. While some focus on supplying as many tools as possible, others focus on the depth of tools. Generally, however, the path taken converges to a single point. There has to be certain fundamental aspects available for this type of software to be viable: access to modules, access to learning materials, access to assignments, access to grades, accessibility, availability and security.

While Eagna will not fill a gap in the market in terms of features or tools, it will provide a unique solution to the university, built on services that already exists. The implementation of the registration system with Brightspace has clearly gone awry, but with the creation of a VLE built on a registration system mimicking the setup for the Technological University of Dublin, there is a viable use-case for this project.

Loop is a prime example of the viability of an in-house design solution. It is specifically designed for and used by Dublin City University. The control over what is implemented, discarding what is unnecessary, and with a grip over cost and scaling, shows that it can and has worked.

Based on the review, the following technologies have been chosen for the development of the Eagna system:

* **Virtual Machines:** Oracle VirtualBox
* **Front-End:** HTML, CSS, JavaScript, HTMX
* **Back-End:** Django
* **Database:** PostgreSQL
* **Monitoring:** Prometheus & Grafana

The virtual machine, front-end, back-end and database stack compliments each individual technological component, with the required features and tools needed to make this project a reality, being available with this stack, with previous examples of usage with one another.

The review of relevant requirements for the implementation of a software system such as this has made clear the true scope of what would be required to implement such a system professionally. When the system specification is being conducted, there will have to be a limited implementation to account for the single developer situation. It is important to include areas such as GDPR and Accessibility in this design.

The review of the previous Final Year Projects will help in understanding the scope and scale required to write a worthy report, while providing a visual inspiration for the implementation of a worthy solution.

# 3. System Analysis

## 3.1 System Overview

This system will be a portal that a student or lecturer can access, to engage in their studies at the Technological University of Dublin. When they access the first page, they will be prompted to either login or register (if they are an unregistered or returning student), and once completed, they will gain access to the main features of the system. Like Brightspace, they can access their modules, with the relevant notes by week, the assignments, the grades, and some other features relevant to studying at TUD.

While the student and lecturer can engage in their studies, there will be a future addition of functionality for system administrators, who will be able to access data, which will be restricted based on legal guidelines, such as General Data Protection Regulation, in order to support the maintenance of the system.

This will include performance metrics relating to the Web Application & Database, access to certain areas of the Web Application for fixing mistakes and errors, alongside certain areas of the Database to assist with misinputs and incorrect data.

In order to fully gauge the needs of the user, the gathering and analysis of the requirements for this integrated Virtual Learning Environment will require the following sections.

## 3.2 Requirements Outline

The key stakeholders for this system would include the following:

* **Students**  
  They would be the most frequent user of the system, and they would depend upon it for their studies.
* **Lecturers**While there would be less lecturers than students, they would also be using this system as frequently as students, if not more, depending on their schedule of modules.
* **Executives**While not a direct user of the system, the implementation of a system such as this would be important to them, as it could affect the reputation of the university, the costs relating to the university and the overall engagement of their hired staff.
* **Maintenance / Security Staff**While also not a direct user of the system, they would have to engage with the operational function of the system, and it would affect their jobs heavily.

To collect the requirements needed for this system to run, the use of knowledge gained from the literature review, research into functional and non-functional requirements, and a collection of user feedback on related systems is needed.

As the only end users of the system that would be engaging in actually using the system, the students and lecturers, they will be the focal point for this requirement gathering, while there will be an implementation of some restricted access for maintenance staff to manage the system, towards the end of the project. The executives, while being end users, do not need to be engaged in this collection of requirements, as their needs could be inferred from what is available in Brightspace, and from the issues the college currently faces.

## 3.3 Questionnaires

For this project, the collection of data from current / past students who have used Virtual Learning Environments, would help to broaden the scope of research and potential analysis of current competitors.

While possessing years of experience using Brightspace and Moodle (*from time spent at Coláiste Dhúlaigh College of Further Education*), everyone’s experience is unique and different, especially for those with learning or physical disabilities.

Accessibility is a primary focus of the web application, and it is essential that modern learning platforms accommodate the needs of the user, whatever they may be.

To help with this, a questionnaire was created for general feedback and for potential accessibility issues, with the aim of delivering it to several different groups of past / present students, a wide array of potential users that would be very helpful in learning from.

Designed to be clear and concise, to maximise the engagement to feedback, it engages with GDPR in the fact that this data would be considered ‘special category’ and that it must be collected based only on the need of the data, that it is kept anonymous, and that the person answering must give their explicit consent in using this data for the project.

As the initial heading and sub-heading, the following was stated:

* **Virtual Learning Environment Survey**
  + This anonymous survey was created to collect feedback from past and current students that have experience using VLE platforms in their studies such as Brightspace, Moodle, Loop, Blackboard or Canvas. This data will be used to help design and implement features for my custom VLE with integrated registration as my final year project for TU856/4 - BSc in Computer Science.

There are three sections for this survey, starting with General Questions:

* **General Questions**
  + What Virtual Learning Environment(s) have you used before? \*  
      
    - Brightspace
    - Moodle
    - Loop
    - Blackboard
    - Canvas
    - *Other (Enter)*
  + How would you rate your experience using the Virtual Learning Environment(s)? \*  
      
    - A scale from 0 to 10 - where 0 means ‘Very Difficult to Use’ and 10 means ‘Very Easy to Use’
  + What is the easiest feature to use in the Virtual Learning Environment(s)? \*  
      
    - 500 Character Limit
  + What is the hardest feature to use in the Virtual Learning Environment(s)? \*  
      
    - 500 Character Limit
  + If there was one thing you could change about the Virtual Learning Environment(s), what would it be? \*  
      
    - 500 Character Limit

The next section is focused on Accessibility, where there was no requirement to answer with any personal details, regardless of the provided anonymity.

* **Accessibility**
  + Do you identify with any of the following disability categories? \*  
      
    - No
    - Yes – Prefer Not To Specify
    - Visual Impairment
    - Hearing Impairment
    - Mobility Impairment
    - Speech Impairment
    - Learning Disability
    - Neurodivergent
    - Long Term Illness
    - Mental Health Condition
    - Other (Specify)
  + How has your relevant disability affected your use of the Virtual Learning Environment?  
      
    - 500 Character Limit
  + What missing feature would help you in using the Virtual Learning Environment?  
      
    - 500 Character Limit

The final section is the implicit request for consent to use this anonymous feedback to help design and improve the project.

* **Consent**
  + Do you consent to me using your anonymous feedback to help design / improve my project? \*  
      
    - Yes
    - No

Alongside this Student Questionnaire, another was created for Lecturers, so that an insight could be gained on the relevant features that they felt were necessary for them to engage with using a Virtual Learning Environment. Using the same layout of questions, but separated into a second form, it was posted to differentiate the data being collected from students. The general and short nature of the questionnaire meant that the questions still apply, and also enable a more engaging response, with no burden to spend more than a few minutes on giving feedback.

## 3.4 Questionnaire Feedback

In this section, a review of the feedback received from the questionnaires will be performed.

Starting off with the student feedback, there was a total of twenty-eight responses, providing plenty of valuable feedback for the student use case scenario.

Regarding the lecturer feedback, there were a lot fewer responses, with the final total being four, but some were received nonetheless, providing valuable feedback for the lecturer use case scenario.

*The base data collected from these questionnaires is available in Appendix A.*

**Student Feedback**

Here is the initial dispersion of VLE usage found in the feedback for the forum, with Brightspace dominating many submissions, with the usage of Moodle and Loop following behind:

A graph with colorful bars

AI-generated content may be incorrect.

Figure 2 - Student Feedback Questionnaire - Which Virtual Learning Environment(s) have you used before?

Here is the overall feeling regarding usage of VLEs with Net Promoter Score, showing generally a negative experience in using these systems:

A screen shot of a device

AI-generated content may be incorrect.

Figure 3 - Student Feedback Questionnaire - General Experience Rating

Much of the student feedback revolves around modules. For both the easiest and hardest elements of using their Virtual Learning Environment, the implementation of modules seems to be the most popular topic, with mixed feelings regarding ease of use, as shown in the following word clouds displayed as data for the form:

A close-up of a computer screen

AI-generated content may be incorrect.

Figure 4 - Student Feedback Questionnaire - Easiest Feature Word Cloud

This is the ‘easiest feature’ feedback, displayed in the word cloud above. Users feel that modules are ‘very approachable’, that ‘browsing modules’ are very easy, and the module content and layout, with a brief mention of assignments, is accessible.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5 - Student Feedback Questionnaire - Easiest Feature Feedback I

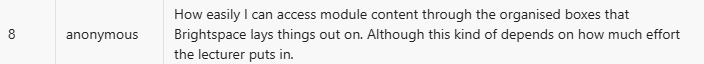


Figure 6 - Student Feedback Questionnaire - Easiest Feature Feedback II



Figure 7 - Student Feedback Questionnaire - Easiest Feature Feedback III

However, not all users felt this way, as will be shown below.

A close up of a computer screen

AI-generated content may be incorrect.

Figure 8 - Student Feedback Questionnaire - Hardest Feature Word Cloud

This is the ‘hardest feature’ feedback, displayed in the word cloud above. The standout point is that lecturer control over the layout of modules, rather than uniformity across them all, causes some students to suffer in their usage. Here are some direct examples:



Figure 9 - Student Feedback Questionnaire - Hardest Feature Feedback I

A close up of text

AI-generated content may be incorrect.

Figure 10 - Student Feedback Questionnaire - Hardest Feature Feedback II



Figure 11 - Student Feedback Questionnaire - Hardest Feature Feedback III



Figure 12 - Student Feedback Questionnaire - Hardest Feature Feedback IV



Figure 13 - Student Feedback Questionnaire - Hardest Feature Feedback V

There are clearly mixed feelings, even with Submitter #8 in the ‘easiest’ feature section (shown by Figure 5).

The main point gathered from this data, is that module access should be easy, but there should be limitations and structure to how learning materials should be laid out in modules.

Other points to note are that the assignment layouts generally are found to be easy, as shown in the feedback below for the ‘easiest’ feature to use:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14 - Student Feedback Questionnaire - Easiest Feature Feedback IV

A screenshot of a computer

AI-generated content may be incorrect.

Figure 15 - Student Feedback Questionnaire - Easiest Feature Feedback V

A screenshot of a cell phone

AI-generated content may be incorrect.

Figure 16 - Student Feedback Questionnaire - Easiest Feature Feedback VI

While some mentioned ‘hardest’ features include enrolment, the Bongo Virtual Classroom, the use of discussions, and the built-in exam / quiz features:



Figure 17 - Student Feedback Questionnaire - Hardest Feature Feedback VI



Figure 18 - Student Feedback Questionnaire - Hardest Feature Feedback VII



Figure 19 - Student Feedback Questionnaire - Hardest Feature Feedback VIII

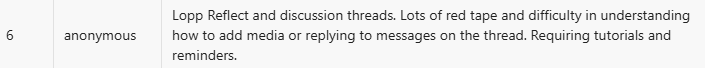


Figure 20 - Student Feedback Questionnaire - Hardest Feature Feedback IX



Figure 21 - Student Feedback Questionnaire - Hardest Feature Feedback X

A close up of text

AI-generated content may be incorrect.

Figure 22 - Student Feedback Questionnaire - Hardest Feature Feedback XI

Once again, when asked what would be the one thing that could be changed, it revolves around the layout of modules, as shown in the word cloud below:

A close-up of a computer screen

AI-generated content may be incorrect.

Figure 23 - Student Feedback Questionnaire - One Thing To Change Word Cloud

The main conclusion gathered from the first part of this feedback, is that modules must be structured better, there is easy access to the main parts of the course (modules, assignments, grades etc.) and that extra tools such as discussions and the Bongo Virtual Classroom are not intuitive and are hard to use.

The second part of the form focused on accessibility, with the following data received:

A graph with different colored bars

AI-generated content may be incorrect.

Figure 24 - Student Feedback Questionnaire - Do you identify with any of the following disability categories?

With submitters identifying as neurodivergent, having a learning disability, and one having a visual impairment, there is some good feedback to be found here. Here are some of the ways in which disabilities have affected the submitters usage of their Virtual Learning Environment:



Figure 25 - Student Feedback Questionnaire - Disability Affected Usage Feedback I



Figure 26 - Student Feedback Questionnaire - Disability Affected Usage Feedback II

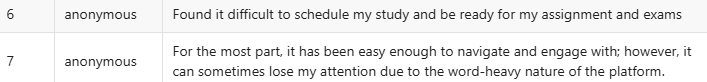


Figure 27 - Student Feedback Questionnaire - Disability Affected Usage Feedback III



Figure 28 - Student Feedback Questionnaire - Disability Affected Usage Feedback IV

Here are some of the suggested missing features (which are feasible) that could help in using the Virtual Learning Environment:



Figure 29 - Student Feedback Questionnaire - Suggested Missing Feature I



Figure 30 - Student Feedback Questionnaire - Suggested Missing Feature II

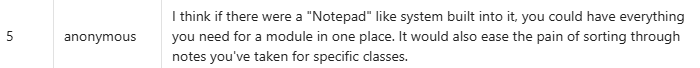


Figure 31 - Student Feedback Questionnaire - Suggested Missing Feature III



Figure 32 - Student Feedback Questionnaire - Suggested Missing Feature IV



Figure 33 - Student Feedback Questionnaire - Suggested Missing Feature V

A screenshot of a phone

AI-generated content may be incorrect.

Figure 34 - Student Feedback Questionnaire - Suggested Missing Feature VI

The conclusion from the second part of feedback is generally related to clarity. There is difficulty found in understanding instructions or in an overload of content thrown at the user.

These can be tackled with an analysis of Nielsen’s Heuristics in the final stages of the project, while options such as Dark Mode (for the UI), support for built-in browser Reading Modes / Text-to-Speech, and a clearer ability through structure to access what they need to engage in their studies.

**Lecturer Feedback**

Here is the initial dispersion of data regarding lecturer usage of various Virtual Learning Environments, with Brightspace once again being the top spot, with Blackboard and Moodle behind.

A graph with different colored bars

AI-generated content may be incorrect.

Figure 35 - Lecturer Feedback Questionnaire - Which Virtual Learning Environment(s) have you used before?

Here is the overall feeling regarding usage of VLEs with Net Promoter Score, showing generally a negative experience in using these systems:

A close-up of a speedometer

AI-generated content may be incorrect.

Figure 36 - Lecturer Feedback Questionnaire - General Experience Rating

With less data to work with, there was no word cloud visual available. For the easiest feature feedback, it was revolving around content upload:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 37 - Lecturer Feedback Questionnaire - Easiest Feature Feedback

For the hardest feature feedback, there were mixed responses, including access to student data (such as who is in their module, and if they should be there), a general overview of grades, management of quizzes and the addition of rubrics and marking schemes:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 38 - Lecturer Feedback Questionnaire - Hardest Feature Feedback

If there was one thing these lecturers could change, it would be the following:

A screenshot of a computer test

AI-generated content may be incorrect.

Figure 39 - Lecturer Feedback Questionnaire - One Thing To Change Feedback

Most of these will not be feasible for the scope of the project, with a deferral to tools such as Excel (for rubric management) and attachment to modules most likely.

In the second part of the form, regarding accessibility, the following data was collected:

A screenshot of a graph

AI-generated content may be incorrect.

Figure 40 - Lecturer Feedback Questionnaire - Associated Disability Category

This shows that only 1 of the 4 lecturers did not associate with a disability category.

The following data displays how their relevant disabilities have affected their use of VLEs:

A screenshot of a phone

AI-generated content may be incorrect.

Figure 41 - Lecturer Feedback Questionnaire - How Associated Disability Category Affects VLE Usage

Regarding this data, there is a similar strain of accessibility that may be needed, with structure, dark mode, and support for text-to-speech that could help the blind users.

Regarding missing features that could potentially help in using the Virtual Learning Environment, here is the following data:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 42 - Lecturer Feedback Questionnaire - One Missing Feature to Help Deal with Disability Category

Regarding this data, the first point may be tackled through the creation of an instructions page, but the aim of this system is to be as intuitive as possible, with features clearly mapped out, so it may not be necessary. A chat / conversation interface, outside of global / module announcements would be outside the scope of this project.

While the feedback here was sparse, and even then, some requested features would be unfeasible, there was some useful information here, relating to complexity. The easiest features were those that were clear and had multiple methods of usage (uploading), while more complex features were harder to use. The scope of this project is to shave off unnecessary features, features that can be utilized through familiar software like Microsoft365, to avoid as much learning as possible (something that will be covered in Nielsen’s Heuristics later).

## 3.5 Brightspace Features Analysis

In this section, there will be an analysis of the features of Brightspace, being the current Virtual Learning Environment of the Technological University of Dublin, which will help in understanding the requirements of this Virtual Learning Environment.

**Homepage**

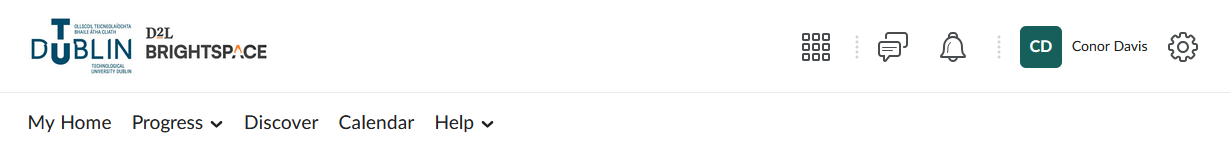


Figure 43 - Brightspace Homepage Header

When you login to Brightspace, you are greeted with the dashboard. At the top of the page is the header, including a navigation menu, the option to search through modules, subscription alerts, notifications, access to the user profile and settings.

A white background with black and white lines

AI-generated content may be incorrect.

Figure 44 - Brightspace Work To Do Widget

There are only a few other features found on the homepage, including the ‘Work To Do’ widget which shows your upcoming, unsubmitted continuous assessment. It is sorted, in descending order, based on the nearest deadline that must be met. If you were to miss out on the submission deadline, a space above is shown for ‘overdue’ submissions, giving them priority at the top to show you have missed it.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 45 - Brightspace Modules Dashboard

Next, is the most important part of the dashboard: modules. Here, you can select any of your enrolled modules, to get to their page. You also have access to extra modules for training, alongside your previous years of study.

**Modules**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 46 - Brightspace Module Header & Navigation Menu

On each module page, you are given the same rough setup as the dashboard, with an expanded navigation menu, to give you quick access to your progress in the module, the module content page, assessment which will give you access to your various types of assignments, module tools, which has a wide array of features including announcements, groups, virtual classroom and more. There is also quick access to the TUD Library page on the main website, along with the help option to gain quick access to documentation information and an FAQ.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 47 - Brightspace Module Learning Materials & Announcements

Next, is an overview of learning material, and announcements. Each module can be unique, as the layout is ultimately determined by the lecturer, and their module needs (according to them). The announcements are module wide, in comparison to the global announcements found on the dashboard.

**Progress**

A screenshot of a progress report

AI-generated content may be incorrect.

Figure 48 - Brightspace Progress Menu

The Progress area of Brightspace gives you a per-module overview of various stats, including your grades, login history, quizzes, assignments, objectives, and login / access history. While some of these statistics aren’t necessarily important to a student at TUD (such as login history or system access history), they may be used for academic integrity reasons by lecturers (such as visiting Brightspace / the learning materials during a quiz).

**Discover**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 49 - Brightspace Discover Module Feature

The discover area of Brightspace is used to discover any module you need to enrol with. You can use the search function to sort by name or module code.

**Other Areas**

Other general areas of Brightspace include the following:

* ***Getting Started***  
  On the dashboard, there is a small section for guidelines and resources to get started with using Brightspace, as an introduction to the features, areas of the website, and tools available to you.
* ***Microsoft365***The TUD version of Brightspace also includes quick access to Microsoft relevant links on the dashboard, including Outlook, Calendar & OneDrive. If you choose to link your account, it will embed your emails and calendar, with no embedding for OneDrive.
* ***Global Announcements***  
  These announcements are available site-wide, and are usually related to Brightspace, or important information regarding TUD.
* ***Calendar***The calendar contains all important dates / times, for your assignments, Bongo Virtual Classes, and any other set dates for assessments (quizzes etc.)

**Lecturers**

While there is no access to the lecturer side of the website, they have an extensive amount of customization available within their direct modules, from how they layout their learning material, creation of assignments, creation of quizzes and more. They largely have access to a very similar user interface / array of features, with the ability to customize their modules in comparison.

## 3.6 Nielsen’s Heuristics

Mentioned in the introduction to this project, Nielsen’s Heuristics have provided the ultimate foundation for web usability for decades [114] and will be important in the future design of this project. Here are the outlined categories based on a recent outline provided by Jakob Nielsen [115]:

* **Visibility of System Status**  
  The user must be provided efficient feedback on their use of the system, so that they can engage in usage properly.
* **Match Between System & Real World**  
  The system must be developed to respect contemporary everyday solutions that are used by the user, in order to make their experience as natural as possible.
* **User Control & Freedom**  
  Ensure the user has the ability to undo their actions, leave their current area of usage.
* **Consistency & Standards**  
  Provide a consistent structure in design, so that the user knows what to expect next intuitively, to make the system easier to use.
* **Error Prevention**  
  Seek to prevent errors where possible and offer the user fair warning in the event of potential consequences.
* **Recognition Rather than Recall**  
  Reduce the need for the user to remember how to use the system, by making their current options visible, not based on past information they have taken in.
* **Flexibility & Efficiency of Use**  
  Ensure that the usage of the system accommodates both the new user, and the expert, focusing on providing an overall efficient and flexible experience that encourages easy navigation.
* **Aesthetic & Minimalist Design**  
  Avoid the implementation of obtuse or unnecessary features and ensure that the implemented design is clear and easily discernible.
* **Help Users Recognize, Diagnose & Recover From Errors**  
  In the event of an error, when error prevention has failed, provide a clear understanding of what has gone wrong, and the next avenue they can take as a solution to their problem, and to avoid further errors.
* **Help & Documentation**  
  Where the design cannot inherently be straightforward, ensure that there is appropriate help and documentation for their navigation of the system.

## 3.7 Requirements Analysis

With the previously researched areas in the literature review, including the alternative existing solutions, technologies and what they will provide, the requirements that could be necessary to implement, an initial systems model shall be developed.

The consensus gathered from the questionnaire relates to usage of modules, complexity of use and a need for structure. While some students were generally happy with how modules were implemented, others were unhappy with the lack of clear structure across them all. The features that were the clearest to use, resulted in being the easiest to use. This was the case for lecturers as well, who were clearly happy with their ability to upload learning materials and attached documents, but struggled with more complex features such as the implementation of rubrics into their modules.

The review of Brightspace showed the currently used features and areas that are being used by the Technological University of Dublin, to give an idea of what features were available on the website, and what should be potentially implemented into this project.

Nielsen’s Heuristics outlines a solution to previously described problems given in user feedback, a standard for the user experience that will help to tackle complexity and ease of use problems, to serve as an evaluation method in the latter stages of the project (post-Interim).

With an idea of how to proceed with the gathered requirements, the design of three use cases, two initial displays for the separate main users, with a final design implementing an expansion of both, and the inclusion of a maintenance administrator, will commence and be shown on the next pages.

The initial use case model for the student consists of the following:

A diagram of a business process

AI-generated content may be incorrect.

Figure 50 - Initial Student Use Case

This use case model depicts the following user flow:

* **1** | The student must login or register to access the system
* **2** | They will then be granted access to the dashboard / homepage
* **3a** | They will be able to use the navigation bar to access the dashboard, relevant tools, email inbox, and the TUD Website.
* **3b** | They will also have access to their modules, upcoming assignments, and recent global announcements.
* **3c** | They will also have a notification bell in the top right corner, along with access to their user profile for personal details.
* **4a** | If they select a module from the Dashboard, they will gain access to the weekly learning material, relevant assignments, and recent module specific announcements created by the lecturer.
* **4b** | If they select the tools from the navigation menu, they will have quick access through links to their available Microsoft365 tools.
* **4c** | If they select Inbox or Website, they will be redirected to their Outlook email or to the official TUD website.
* **4d** | If they select their user in the top right, they will gain access to personal details.

The initial use case for the lecturer will consist of the following:

A diagram of a software system

AI-generated content may be incorrect.

Figure 51 - Initial Lecturer Use Case

This use case model depicts the following user flow:

* **1** | The lecturer must login to access the system
* **2** | They will then be granted access to the dashboard / homepage
* **3a** | They will be able to use the navigation bar to access the dashboard, relevant tools, email inbox, and the TUD Website.
* **3b** | They will also have access to their modules, submissions not yet graded, and recent global announcements.
* **3c** | They will also have a notification bell in the top right corner, along with access to their user profile for personal details.
* **4a** | If they select a module from the Dashboard, they will gain access to the weekly learning material to edit, relevant assignments, and module specific recent announcements created by them.
* **4b** | If they select the tools from the navigation menu, they will have quick access through links to their available Microsoft365 tools.
* **4c** | If they select Inbox or Website, they will be redirected to their Outlook email or to the official TUD website.
* **4d** | If they select their user in the top right, they will gain access to personal details.

The final use case, showcasing implementation of student, lecturer and administrator needs is shown as follows:

A diagram of a company

AI-generated content may be incorrect.

Figure 52 - Expanded Use Case

This use case model depicts the following user flow, with the following formatting:

* **1-S-a** = 1st Step, Student Flow, Option A
* **1-L-a** = 1st Step, Lecturer Flow, Option A
* **1-A-a** = 1st Step, Admin Flow, Option A
* **1-S-a** | The student must login or register to access the system
* **2-S-a** | The student is then shown the dashboard, with access to the navigation menus, module selection, assignments, recent global announcements, and their user profile.
* **3-S-a** | If the student selects a module, they will gain access to recent module announcements, their academic calendar and module assignments.  
  + **4-S-a** | If the student selects a module assignment, they will be re-directed to the assignment page with submissions. Here, they will have access to their grades, the attached assignment files and all necessary information included by the lecturer (due date / time etc.)
  + **4-S-b** | If the student selects a file from their academic calendar, the file will be downloaded or displayed.
  + **4-S-c** | The student can view recent module announcements.
* **3-S-b** | If the student uses the navigation menus, they will be re-directed to the TUD website, their TUD Email, or to a quick access page for Microsoft365 Tools.
* **3-S-c** | If the student selects an assignment, they will be re-directed to the assignment page with submissions. Here, they will have access to their grade, the attached assignment files and all necessary information included by the lecturer (due date / time etc.)
* **3-S-d** | If the student clicks their name, they will be given access to their user profile, containing their details (Name, Email, Course, Year, Modules, Overall Percentage)
* **3-S-e** | The student can view recent global announcements.
* **1-L-a** | The lecturer must login to access the system
* **2-L-a** | The lecturer is then shown the dashboard, with access to the navigation menus, module selection, assignments, recent global announcements, and their user profile.
* **3-L-a** | If the lecturer selects a module, they will gain access to recent module announcements, the academic calendar and module assignments.  
  + **4-L-a** | If the lecturer selects a module assignment, they will be re-directed to the assignment page with submissions. Here, they will have access to grading, the attached assignment files and all necessary information included during creation (due date / time etc.).
  + **4-L-b** | The lecturer can view all possible weeks in the academic calendar, edit the description for that week, and attach / remove learning materials.
  + **4-L-c** | The lecturer can create module announcements.
  + **4-L-d** | The lecturer can create / delete assignments, leading to a small page to set a title, description, due date / time, the overall mark of that module in relation to the assignment, along with attaching relevant documents.
* **3-L-b** | If the lecturer uses the navigation menus, they will be re-directed to the TUD website, their TUD Email, or to a quick access page for Microsoft365 Tools.
* **3-L-c** | If the lecturer selects an assignment, they will be re-directed to the assignment page with submissions (all submissions for that assignment). Here, they will have access to grading, the attached assignment files and all necessary information included during creation (due date / time etc.).
* **3-L-d** | If the lecturer clicks their name, they will be given access to their user profile, containing their details (Name, Email, Year, Modules, Total Grade Submissions ../..)
* **3-L-e** | The lecturer can view recent global announcements.
* **1-A-a** | The admin must login to access the system.
* **2-A-a** | The admin is then shown their admin panel, with access to adding lecturer accounts, editing student enrolment, editing lecturer enrolment and posting global announcements.

## 3.8 Initial System Specification

**Functional Requirements**

Here are the initial functional requirements of the Eagna system, composed of the web application, database, monitoring platform, and infrastructure requirements, with each interlocked, and dependent on one another.

**Web Application**

* **Login & Registration System**

The user, student or lecturer, should be able to use their email address to login, along with their associated email password. The student should also be able to enrol in modules through registration.

* **Module Access**  
    
  The user, student and lecturer, should be able to access their relevant modules in this system.
* **Learning Material Access**  
    
  The user, student and lecturer, should be able to access / upload the learning material (respectively) once they have accessed the relevant module.
* **Assignment Access**  
    
  The user, student and lecturer, should be able access & submit / access & grade the relevant assignment inside the relevant module.
* **Recent Announcement Access**  
  The user, student and lecturer, should have access to recent global announcements created by an admin, or the student should have access to recent module announcements created by the lecturer.
* **Extra Tools Access**  
    
  External tools, such as quick access to the TUD website, Microsoft Word or Microsoft Teams through Microsoft365, should be provided.

***Database***

* The design of a schema necessary to support the functionality of the web application should be implemented.
* The design should also account for non-functional requirements (such as Security & GDPR) where necessary.

***Monitoring Platform***

* The monitoring platform should give maintenance staff access to availability metrics to determine the overall health and status of the system.
* The monitoring platform should give maintenance staff access to performance metrics to determine the overall speed and usability of the system.

***Infrastructure***

* For this system to work, the various nodes (virtual machines) should be deployed in a network where they have access to one another for the system to function.

**Non-Functional Requirements**

Here are the initial non-functional requirements of the Eagna system, decided upon and verified through research conducted, an evaluation of Nielsen’s Heuristics, and feedback gathered through the questionnaire.

* ***Availability***  
  The design principles of **Redundancy** will be ***included***, by the nature of the system design. This is achieved by having each integral part of the system isolated to separate environments, whereby they can be duplicated and made available in a theoretical real-world deployment. The design principles of **Observability** will be ***included***, where there will be a typical expected input and output for the system, such as entering login details and gaining access to the system, and a failure in that area can be inferred to be related to those individual files, or a database failure. The usage of a monitoring system will also be able to give an inferable insight through metrics. The design principles of **Recovery** will be ***excluded***, as this will generally just come through general maintenance of the software in a theoretical real-world deployment, which cannot be wholly accounted for.
* ***Performance*  
  Caching** is natively supported with Django and PostgreSQL combined, so it is unlikely that a further implementation (through the likes of Redis) will be included. **Concurrency** is also supported with the core technologies with Django orchestrating web requests and access to the database, while PostgreSQL enforces concurrent access rules, giving native compatibility. For **Query Optimization**, this will be performed in the latter stages of the project, when all queries necessary have been implemented [116].
* ***Security*  
  Authentication** is natively supported by Django, as previously reviewed during the Literature Review, which will be included in the design and implementation for this system to handle the users. **Encryption** will be considered in the latter stages of the project, should there be space to implement it, through the usage of TLS to secure Django, with PostgreSQL able to support encrypted data storage already. **Access Control** will be integral to the program design, with a login required to access the system, alongside a separation of access between the functionality of a student and a lecturer.
* ***GDPR***  
  As according to the **GDPR** legislation, only the necessary data for the students will be included. Their names, email addresses, passwords and course relevant information, such as grades, modules and assignments, will be stored in the database, with attached files being stored directly on the drive, to support performance, as they are required for the functionality of the application. Only parts of this data will be accessible to administrators, for functionality purposes (such as course relevant information if they are to be added / removed from modules).
* ***Accessibility***

Some of the requests from users in the feedback questionnaire will be implemented based on feasibility, such as Dark Mode, while there will be the aim to provide accessibility through compatibility with built-in text-to-speech found in browsers with reading modes. There will also be potentially a review of the following document [117], which contains UI design outlines for various neurodivergences and learning disabilities, such as autism and dyslexia, in tandem with a review of Nielsen’s Heuristics. These posters can be found in the 2nd Floor TUD Lab Rooms.

**The Architecture**

The appropriate architecture for this system, based on the outlined review and analysis, is the deployment of a distributed system, containing a separation between the web application, database and monitoring system, while remaining interconnected. This will be done using virtual machines, which in practice, work together as a distributed system, providing a development environment to ensure functionality and connection, and future support for deployment on various servers, or into the cloud.

## 3.9 Conclusions

The analysis has resulted in a great many requirements necessary for a worthy system to be implemented as an actual Virtual Learning Environment. While the aim is to try and achieve the most usable and accessible environment possible, it will not be possible to cater to every potential need.

While there are four listed stakeholders, only students and lecturers will be the main focus, while still providing consideration for maintenance staff through the later deployment of a monitoring system and administration panel in the web application.

The questionnaire provided reasonable feedback into what was needed, while the Brightspace review gave an idea of the main functionality used by the Technological University of Dublin.

Nielsen’s Heuristics will provide a good foundation for later UI analysis, while the use case models, functional and non-functional requirements, provide the foundation for the start of the design process, found in the next section.

# 4. System Design

## 4.1 Introduction

This section focuses on the design of the system, featuring a review of software methodologies, including the plan outlined with the chosen methodology, an overview of the system (architecture and infrastructure), the system design, and a brief overview of the prototype design.

## 4.2 Software Methodology

For the choice of an appropriate methodology to develop the system, it would be a Feature Driven Development implementation, derived for the fact that a single developer will be working on this project.

Feature Driven Development is an iterative software methodology that focuses on the development of a single functioning piece of your software, at a time. When it has been completed, the next block is tackled [118]. Once the features necessary to be developed have been outlined, work can begin. It is important to focus on the priority of each feature, tackling the most important one first.

While it is typically used as an Agile Framework [119], which is not appropriate for use by a single developer, the core nature of Feature Driven Development, with an outline slimmed to fit a single developer, provides the perfect foundation for developing this system.

The pathway to ensuring this system is working correctly, starting at the access of the system (login), and ending with access to deeper features (like accessing modules notes), means there is a clear starting point A, and a clear ending point B, that can be worked towards incrementally with this framework as a basis.

The design of this system is inherently suitable for this approach to software development, given that you need login access to modules, module access to notes and assignments, assignment access to submission and grades. If the user cannot login, then there is no point in tackling features that they will not be able to access.

Starting with the development of the overall model, followed by the development of the features list, planning out of the features list (and their priority of implementation), and then the design and implementation of the features, this will be a suitable choice [118] [120].

The outline for using Feature Driven Development is as follows:

* **Develop The Overall Model**  
  In this step, the creation of the overall outline of the project is done, whereby the scope of the system is defined, before further details are developed upon in later steps. The background for this step was performed throughout the project up until this point, based on research and requirements gathering, such as what is included in a typical Virtual Learning Environment, in order to develop the overall model, which will be given an overview in the next section.
* **Build A Features List**  
  In this step, the features of the system are outlined, based on our previous research and requirements gathering. If a feature will take too long to create, it should be broken down further into multiple features. The features must fit into the overall scope of the project. Where it has been previously defined, that the inclusion of Microsoft365 tools will be accounted for, there is no point in attempting to build features that will overlap with their inclusion, therefore pushing beyond the scope of the overall model.
* **Plan By Feature**  
  In this step, the priority of each feature, from the built feature list, is determined. In this case, the most important features based on priority will be implemented within the Interim Prototype, with other features and development requirements to be implemented later in the schedule. As an example, the priority of the login system, in order for users to access the system, is paramount, whereas registration for students, will not be necessary until later on, in regard to functionality priority, as students can be created outside this feature, while implementation is taking place.
* **Design By Feature**  
  In this step, based on the priority outlined in the previous step, features will be designed and implemented. For example, for the web application, wireframes will be constructed for the user interface, in the case of the database, a rough database schema, converted into an Entity Relationship Diagram will be created, and in the case of the monitoring application, the potentially important metrics relating to a system such as this will be outlined.
* **Build By Feature**  
  In the final step, the implementation of the designed feature goes ahead. Once a working prototype of the feature has been evaluated and tested, it can be added to the main build. In the case of programming for the web application, an initial skeleton will be generated for the feature by prompting ChatGPT, before implementation into the development environment on the Virtual Machine, with editing and testing taking place in order to confirm proper functionality.

Other methodologies that were researched and then discarded include the following:

* **Test Driven Development**This methodology is an Agile software development-based approach on comprehensive testing, whereby unit tests are defined before any implementation of production. It follows three short cycles, known as red, green and refactoring, with the first focusing on writing a unit test for unimplemented functionality, the second focusing on delivering code to pass the red unit test, and final focusing on refactoring code where necessary while constantly testing.  
    
  The advantage of Test Driven Development is the thorough approach to testing, which will help ensure valid functionality throughout the development process, by catching errors as early as possible in development.  
    
  The disadvantage of Test Driven Development is also the thorough approach to testing, which will hamper speed of development while trying to meet unit tests that will not be found extensively throughout this project, rather only for the core functionality. This project is a web application, with a heavy focus on backend complexity linked to a database, so it would be better to focus on testing how data is moving and connecting to the database, rather than extensively using unit tests for the integrity of the Django / Python code.  
    
  [121]
* **Rapid Application Development**This methodology is an Agile software development-based approach on speed and flexibility, whereby focus is given to rapidly creating prototypes and usable functionality, over fully planning out the project before diving into implementation. The project scope would be outlined, followed by a design phase that includes the creation, testing and refinement of prototypes, before final production and implementation phases.  
    
  The advantage of Rapid Application Development is in the name, in that it used to rapidly develop applications, which would be helpful in developing this project, given the short time frame between the start of the project and when it is due.  
    
  The disadvantage of Rapid Application Development, however, is that it focuses on the development of systems with a light focus on back end complexity and a high focus on front end interactivity, which is the inverse of the objective of this project.  
    
  [122]

## 4.3 The Overall Model

The logical architecture of Eagna consists of the following layers, isolated within their own virtual machine, designed to connect and interact with one another:

**The Web Application Environment**

* ***Presentation Layer***  
  This is front-end of the Eagna web application, consisting of the use of *HTML*, *CSS* & *HTMX* (*JS*), supported by the functionality of the application and data layers.
* ***Application Layer***  
  This is the back-end of the Eagna web application, consisting of the use of *Django*, to interact between the presentation and data layers.

**The Database Environment**

* ***Data Layer***This is the database functionality of Eagna, consisting of the use of *PostgreSQL*, with pgadmin4 for ease of development. This layer interacts with the application layer.

**The Monitoring Environment**

* ***Observability Layer***  
  This is the monitoring system for Eagna, consisting of the use of Prometheus and Grafana to pull and view metrics from each component of the system

A diagram of a software development network

AI-generated content may be incorrect.

Figure 53 - Eagna System Deployment

The overview of the Eagna system concludes the development of the overall model, in relation to the Feature Driven Development lifecycle. It features the use of all core technologies (those outlined in the chosen stack), alongside considering what will be necessary to support the various parts of research, analysis and requirements gathering conducted in the earlier stages of the report.

## 4.4 The Features List

The second step of the Feature Driven Development lifecycle will be outlined within this section, whereby the features list is built.

**Features List**

Based on the gathered research and analysis, such as the functional requirements, here is the features list for the web application.

* **Login System**  
  Students and Lecturers should be able to use their email address to login, along with their associated email password.
* **Registration System**

Students should be able to register, entering their name, email address, password, course and their choice of modules.

* **Dashboard**The user, student and lecturer, should gain access to their relevant dashboard upon logging in, whereby the student can view assignments, modules, recent global announcements, with access to navigation menus in the footer and header, alongside notifications and their user profile, while the lecturer can view ungraded submissions by assignments, their modules, recent global announcements, alongside access to navigation menus in the footer and header, with notifications and their user profile.
* **Modules**  
  The user, student and lecturer, should be able access their relevant modules, where they can view relevant assignments, learning material and recent module announcements. For features regarding students, they should be able to click on their assignments for further details, alongside accessing attached files uploaded by the lecturer according to the academic calendar and relevant assignment. For features regarding lecturers, they should be able to create new assignments, view their list of students, and remove anything no longer necessary (assignments miscreated, notes unintentionally uploaded etc.), alongside creating announcements.
* **Assignments**  
  The user, student and lecturer, should be able access to access specific assignments either from their dashboard or the related module. Once accessed, the student should be able to view all necessary details, associated files, grading, and the submission deadline, while also able to create submissions with an attachment of uploaded files. For the lecturer, they should be able to create assignments from the module page, view all student submissions, and have the ability to grade them / provide feedback.
* **Recent Announcements**The user, student and lecturer, should have access to global announcements created by an administrator, or the student should have access to module announcements created by the lecturer. Viewing will be based on most recent in creation.
* **Extra Tools Access**  
  External tools, such as quick access to the TUD website, Microsoft Word or Microsoft Teams through Microsoft365, should be provided through links in the navigation menus.
* **User Profile**  
  The user, student and lecturer, should have access to a user profile displaying their relevant personal information, such as their names and emails, with students seeing their course, modules enrolled, alongside their grades per module. For lecturers, they should see the modules they are running, alongside the total amount of submissions they have graded out how many have been submitted to them.
* **Administrator Panel**Solely for maintenance purposes for administrative staff, this panel should allow the addition of new lecturers to the system, new modules to the system, editing of student enrolment (addition / removal), editing of lecturer enrolment (addition / removal) and the posting of global announcements.
* **Notifications**  
  The user, student and lecturer, will receive relevant notifications during usage of the platform, such as upcoming deadlines (24 Hours / 1 Week) and released grades for students, and new submissions for lecturers to grade.
* **Accessibility Considerations**Implementation of various accessibility features will be included, such as a variety of colours schemes to account for visual impairments, such as colour blindness. There will also be a design consideration for reading mode / text-to-speech browser features to accommodate a variety of learning difficulties and disabilities. Various parts of user feedback have been considered, such as structure to the module pages for consistency across modules.
* **Monitoring Platform**  
  Using Prometheus and Grafana, maintenance staff should be able to scrape relevant metrics with Prometheus, while visualizing them with the Grafana dashboard.

## 4.5 Plan By Feature

The third step of Feature Driven Development is planning by feature, in order to determine the priority of each feature for implementation. Priority will be given according to dependency, risks to overall system functionality and workload. The implementation of features for the Interim Prototype will be based on this list, with the eventual implementation of all being planned in the final iteration, based on relevance, in case there is issues in implementation later down the road.

1. **Login System**  
   The first feature planned for implementation, this will form the basis for authentication, access control and user access to the system. The whole system depends on the correct user gaining access to their relevant features, and while it will not be the heaviest in terms of workload to implement, all other features will need this working before further implementation can occur.
2. **Dashboard**The second feature planned for implementation, this is the core of the functionality of the web application, whereby the user will be able to access all that they need. Here, they will gain access to navigation menus, assignment information, modules, and every listed piece of functionality listed afterwards. It is the biggest dependency outside of system access through login, and it must work for the rest of the system to be accessed. It will probably be the heaviest in workload.
3. **Modules**  
   The third feature planned for implementation, this, alongside assignments, forms the core engagement with studies for the Virtual Learning Environment. From this feature, the user will have access to learning materials, the relevant assignments, and all the important aspects of course engagement. Various parts of the system, such as modules, announcements, the registration system, and the user profile, all depend on modules being correctly implemented, making it risky to not have implemented when developing the following features, as a dependency for all these other features.
4. **Assignments**  
   The fourth feature planned for implementation, this, alongside modules, forms the core engagement with studies for the virtual learning environment. From this feature, students will be able to access their continuous assessment, the relevant files relating to it, how it impacts their overall grade in the module, alongside uploading their own submissions, while lecturers can create these assignments, set the overall mark, a description, attach relevant materials such as descriptors or rubrics, and then view all submissions for grading. An important part of the system, it will be implemented into the dashboard, into each module, weighing it lower compared to those dependencies, with priority over the following features due to how it links into the system.  
     
   ***These first four parts are the minimum requirements for the system to be operational, and so they will be the priority for the interim for implementation. They are the heaviest in terms of workload, and if missing or broken, present a serious risk for the project, so it is best to implement them as early as possible.***
5. **Registration System**

The fifth feature planned for implementation, and the first following the Interim, it will allow students coming to the system in a new year, or into the college for the first time, to enrol in their modules, with access through the login screen. As one of the proposed benefits of this system for the college, it is first in priority to be implemented post interim as a non-dependent part of system functionality. It is relatively lightweight in workload, linking to the system in the same way as the login system, with the addition of taking in user input for user creation.

1. **User Profile**  
   The sixth feature planned for implementation. Some of the users in feedback requested an easier overview of how they are doing, and this will be done here with marks for individual modules. It will be relatively lightweight to implement, with edits to the user implementation, and querying for pulling details and module marks. Some features listed before this point are dependencies, pushing it down the list.
2. **Recent Announcements**The seventh feature planned for implementation. It will allow for the viewing of recent global announcements on the dashboard, and recent module announcements. The global announcements creation will be done by admins later in the administrator panel, while module announcements can be created by lecturers. Some of the features listed before this point are dependencies and while it will be heavier in workload than user profiles and registration, it is not as big of a priority for implementation.
3. **Notifications**  
   The eight feature planned for implementation, it will notify the user in regard to important details, such as upcoming deadlines for assignments or new submissions for assignments by students. Relatively lightweight as well, some of the features listed before this point are dependencies, pushing it further down the list.
4. **Extra Tools Access**  
   The ninth feature planned for implementation; it will be a simple webpage with hotlinks to Microsoft365 tools available as a student of TUD. Light in workload and easily accessible through the navigation menu, it will not take long to implement.
5. **Accessibility Considerations**The tenth feature planned for implementation, it is relatively important, but requires being further down in the list, in order to cover everything implemented before this point. While not strictly risky in terms of functionality, it is important, and will require a decent workload, where it has not already been accounted for.
6. **Administrator Panel**The eleventh feature planned for implementation; it is important for administrators but is not really relevant to the vital end users – the student and lecturer. It will be similar in workload to Extra Tools, but heavier as it links to Django and the database.
7. **Monitoring Platform**  
   The final feature planned for implementation; it will be accessible to relevant staff through the browser. It will require a decent workload but is not a necessity for system functionality, and it would be best to implement after all other features.

## 4.6 Design By Feature

Through the use of feature-driven development, here is the proposed design outline for the system. In this step, the designing by feature process will take place, based on the built features list that will need to be implemented for Eagna. It will primarily be handled through wireframes for the web application, with the database necessary to support the vast majority of the functionality, which will be presented as an Entity Relationship Diagram, as an overview for all features. The monitoring platform will also have its own section, as the last feature to be designed and built.

**Web Application**

The essential core of the Eagna system, the interface design will be shown here, based on the plan for features previously mentioned, through wireframes.

***Due to the number of wireframes that have been created, they have been placed in the Design Appendix, which is D, in order based on the plan for features listing.***

An example of the styling of the wireframes can be found below:

A blue rectangular box with white text

AI-generated content may be incorrect.

Figure 54 - Wireframes - Notification Menu (Click Bell)

For accessibility features later down the line, there will be an inclusion of colour palette options, based on data found here [123], and while blue schemes are the best all-around option for helping people with colour blindness, it is best to leave them with some options. This will also include our dark / light mode schemes (with a dark variant now the default):

A blue rectangle with white circles and orange circles

AI-generated content may be incorrect.

Figure 55 - Colour Scheme Picker

The particular displayed options above, from left to right, are:

* ***Dark Mode – Default***  
  Mix of desaturated lighter and darker blues, fitting a Dublin / TUD colour scheme.
* ***Light Mode***  
  Mix of white and the lighter blue, another fitting Dublin / TUD colour scheme.
* ***Tritanopia***  
  The only known colour-blindness that affects the perception of blue (blue-yellow deficiency), purple to orange is perceptible to those with Tritanopia.
* ***Achromatopsia***  
  With close to no ability to perceive colour, it is best to have a white / black high contrast mode so there is no trouble in perceiving the interface.

Alongside this colour palette, there will be accommodation for text-to-speech readers, whereby hidden paragraphs may be created as explanations, that can be read by these readers but are invisible on the user interface.

To top off the accessibility implementation, the usage of the following reference [117], used by posters found inside some of the lab rooms on the second floor of the Grangegorman campus for Accessibility Design, will be included where possible. An example of this would be the inclusion of aligning text to the left, with a consistent layout for dyslexia users, while the manual production of audio files will be difficult (and not necessarily needed if text-to-speech accessibility features such as ‘Reading Mode’ on Chrome are accounted for).

The usage of the following design patterns will help in creating an organized codebase for the application:

***Decorator Patterns***

The concept of this pattern is to create an object, or function, that features behaviour used throughout the web application.

The main implementation of this pattern will be @login\_required [124], which will wrap various functions to ensure that the user accessing the functionality is logged in, using Django’s built-in authentication, with a re-direction to the login page should the requirements not be met.

Another implementation of this pattern will be @require\_http\_methods [125], which will be used to manage HTTP methods, such as GET & POST.

***Factory Patterns***

The concept of this pattern is for object creation to be centralized, in order to avoid multiple constructors calls everywhere throughout the system code.

The main implementation of this pattern in this system will be how Django handles object creation through methods, such as the create\_user method for the User class [126]. It will handle password hashing, settings defaults for unfilled entries, and will ensure consistency during the creation of these objects.

***Adapter Patterns***

The concept of this pattern is for enabling of two incompatible interfaces to work with one another using a translation layer.

The main implementation of this will be in the backend settings regarding database usage, where Django is database agnostic. In the settings.py file, the engine used (such as SQLite or PostgreSQL) can be interchangeable, depending on your needs [127]. All it requires is another migration to be run, and new data to be inputted.

These are the primary examples of how Django uses design patterns in order to maintain your codebase.

**Database**

For the web application to function as expected, the implementation of a database is essential.

While an Entity Relationship Diagram has been created, to give a rough outline for how the database will store relevant information, the physical creation of the database schema will be handled by Django.

This will be done through Django’s Migrations, which will automatically map implemented code into the chosen database schema, which in this case, is PostgreSQL.

Here is the initial Entity Relationship Diagram as an outline:

**

Figure 56 - Entity Relationship Diagram

Where this ERD will differ to the actual schema implementation, is in how Django chooses to implement its own tables to handle functions such as User Authentication and Schema Creation. It will automatically create additional tables to handle this process.

Some of these tables will include:

* **accounts\_user\_groups**  
  This will be used to link user entries to Django groups.
* **accounts\_user\_user\_permissions**This will be used to link user entries to Django authentication permissions.
* **auth\_group**  
  Used for Django authorization groups.
* **auth\_group\_permissions**  
  Used to link authorization groups to permissions.
* **django\_admin\_log**  
  Used to record changes made by Django administrators, such as using CRUD functionality.
* **django\_content\_type**  
  Used to store different models used throughout the project, such as a Module.
* **django\_migrations**  
  Used to track Django Migrations to the database, so that if a new migration needs to be run when schema changes are required, it knows what is already done and can avoid re-running it.
* **django\_session**  
  Used to store session data for logged-in users and is important to Django’s authentication.

Below is an example flow of how the system could access usernames (student / lecturer email address) in the database, from user input on the webpage login form, until it tries to query for a valid record in the PostgreSQL database:

A diagram of a computer

AI-generated content may be incorrect.

Figure 57 - System Flow Example

**Monitoring Platform**

While the monitoring platform is not to be implemented until post-Interim, it is important to outline how it shall be implemented.

The third virtual machine will contain the monitoring platform, which will utilize Prometheus and Grafana.

Prometheus would scrape data from each virtual machine using exporters, such as node\_exporter (CPU, RAM, Disk, Network Metrics) and django-prometheus (Application Metrics) setup in the Web Application virtual machine, and node\_exporter and postgres\_exporter (Database Metrics) on the Database virtual machine.

The node exporter is usually exposed on port 9100 on both, with a metrics endpoint added to the web application, and Postgres exporter exposed on a different port, such as 9168.

For the visualization in Grafana, the following metrics have been considered for implementation:

* ***Infrastructure***
  + CPU Usage
  + Memory Usage
  + Disk Usage
  + Network Traffic
* ***Application***
  + HTTP Request Rates
  + HTTP Errors Rates
  + Current Logins
* ***Database***
  + Database Status
  + Active Connections
  + Query Performance

## 4.7 Conclusions

In conclusion, the usage of Feature Driven Development was chosen due to the feature-by-feature nature of the development, which suits the concept of this project. This is because there are multiple dependencies that rely on one another for functionality, leading to a flow of one feature into another.

With roughly twelve planned features in this design, each part has been laid out with wireframes, diagrams and explanations for usage and implementation, based on previous research and analysis.

# 5. Testing and Evaluation

## 5.1 Introduction

In this section, there will be an outline of the testing and evaluation approaches that will be utilized towards the end of the project. While brief testing will take place throughout the project, in order to ensure functionality of the prototype and beyond, the thorough testing and evaluation will take place after the full development of the system.

## 5.2 Plan for Testing

**Web Application Testing**

The testing methodology that will be used for the web application will be ***Black Box Testing***, which will be used to test the functionality of the system features. Django’s implementation of the standard Python Unit Testing framework shall also be used in the end stages of the project. In order to slim down parts of the testing, functionality such as Extra Tools Access will not be explicitly outlined for the lecturer, as such features, when they work for one user, will work for all with access.

***Outlined Structure***

*Student User*

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Functionality | Steps | Expected Results |
| W-S-01 | Login | * Start At Login Page. * Enter Login Details. * Click Login. | User gains access to Dashboard. |
| W-S-02 | Module Access | * Start At Dashboard. * Click on Module. | User gains access to Module. |
| W-S-03 | Assignment Access | * Start At Dashboard. * Click on Assignment. | User gains access to Assignment. |
| W-S-04 | Learning Material Access | * Start At Dashboard. * Click on Module. * Scroll to Academic Calendar. * Click any Attached File. | The User downloads the file or views it within their browser depending on format. |
| W-S-05 | Assignment Material Access | * Start At Dashboard. * Click on Assignment from Dashboard. * Click any attached file for the Assignment. | The User downloads the file or views it within their browser depending on format. |
| W-S-06 | Navigation Menu | * Start At Dashboard. * Click On Navigation Menu Item. | The user should be directed to the correct linked page. |
| W-S-07 | Create Assignment Submission | * Start From Module. * Click On Assignment. * Upload File for Submission. * Submit. | The submission is viewable afterwards on the same page. |
| W-S-08 | View Grade | * Start From Module. * Click on Assignment where Submission has been made. | Any graded submissions should be viewable. |
| W-S-08 | Register | * Start At Login Page. * Click Register. * Fill Out Details. * Finish Registration. | The user should gain access to the system with their new account. |
| W-S-09 | View User Profile | * Start At Dashboard. * Click on Name in Header. | The user should gain access to their user profile. |
| W-S-10 | View Recent Announcements | * Start At Dashboard. * View Recent Global Announcement. * Go To A Module. * View Recent Module Announcements. | The user should be able to see the three most recent global and module announcements. |
| W-S-11 | Notifications | * Start At Dashboard * Click on Bell In Header. | User should see the notifications submenu appear. |
| W-S-12 | Extra Tools Access | * Start At Dashboard. * Click on ‘Tools’ in Navigation Menu. * Click On Each Available Item. | User should be directed to the correct linked tool. |

*Lecturer User*

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Functionality | Steps | Expected Results |
| W-L-01 | Login | * Start At Login Page. * Enter Login Details. * Click Login. | User gains access to Dashboard. |
| W-L-02 | Module Access | * Start At Dashboard * Click on Module. | User gains access to Module. |
| W-L-03 | Assignment Access | * Start At Dashboard * Click on Assignment. | User gains access to Assignment. |
| W-L-04 | Academic Calendar Editing | * Start At Dashboard * Click On Module * Go to any week in the Academic Calendar. * Update Description. * Attach File. | Student users should be able to gain access and view updated description / attached files. |
| W-L-05 | Create Assignment | * Start At Dashboard * Click Module * Click Create Assignment * Fill Out Details * Submit | A new assignment should be visible within the module page. |
| W-L-06 | Add Grade | * Start At Dashboard * Click on assignment where student has submitted. * Click Grade Button. * Fill out details. | Grade should now be available on submission. |
| W-L-07 | Create Module Announcement | * Start At Dashboard. * Go To A Module. * Create Module Announcement. | A new announcement should be visible within the module page. |

*Admin User*

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Functionality | Steps | Expected Results |
| W-A-01 | Login | * Start At Login Page. * Enter Login Details. * Click Login. | User gains access to Admin Panel. |
| W-A-02 | Add Lecturer | * Start At Admin Panel. * Click Add Lecturer. * Fill Out Details. * Submit. | Lecturer should now be able to access system with default password ‘devpass’ |
| W-A-03 | Add Module | * Start At Admin Panel. * Click Add Module. * Fill Out Details. * Submit. | Module should now be available in system and visible when enrolled upon and during registration. |
| W-A-04 | Edit Enrolment | * Start At Admin Panel. * Click Edit Enrolment. * Search By Email For User. * Remove From Module. * Add Another Module. * Submit. | User should now see new module in their dashboard for selection, and the removed module is no longer there. |
| W-A-05 | Create Global Announcement | * Start At Admin Panel. * Click Post Global Announcement. * Fill Out Details. * Submit. | Users should now see a global announcement on their dashboards. |

*Unit Testing*

When all code has been implemented towards the completion of the project, tests will be run using Django’s unit testing framework (built on Python’s unit test framework unittest):

* **Model Unit Test**  
  This test case will be used on the built-in models for users, and how the relationships between them behave. It will be used to make sure the likes of the is\_student() and is\_lecturer() helpers are behaving as expected.
* **Authentication Unit Test**  
  This test case will be used to simulate browser POSTs to the login functionality to verify the login cases of users.
* **Access Control Unit Test**This test case will be used to simulate access control, whereby a student cannot access a lecturer only view, and lecturers have GET access for assignment forms, and POST for creation.

**Database Testing**

The testing methodology that will be used for the database will be ***White Box Testing***. The ways in which database functionality will affect the end user will not be visible, therefore it is best to approach this part from an internal perspective. This will be done by viewing the data stored in the database, and if that aligns with what is presented to the user.

***Outlined Structure***

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Case | Steps | Expected Results |
| D-A-01 | Django Migration | * Perform migration with manage.py | Schema should be visible in pgadmin4. |
| D-A-02 | Data Injection | * Create New Assignment as Lecturer User | Data should be visible in pgadmin4. |
| D-A-03 | Registration | * Register a new user. | Data should be visible in pgadmin4. |

**Monitoring Platform Testing**

The testing methodology that will be used for the monitoring platform will be ***White Box Testing***. The typical end user will never view this part of the system. Actions will be performed through the web application to view data within the Grafana dashboard.

***Outlined Structure***

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Case | Steps | Expected Results |
| M-A-01 | Infrastructure Monitoring | * Login as a student and as a lecturer. * View Grafana Dashboard. | Change in Network Traffic Metrics should be viewable. |
| M-A-02 | Database Monitoring | * Visit Module as Lecturer. * View Grafana Dashboard. | Change in Query Performance Metrics should be viewable. |
| M-A-03 | Application Monitoring | * Login as a user. * View Grafana Dashboard. | Change in Active Logins Metrics should be viewable. |

## 5.3 Plan for Evaluation

**Web Application Evaluation**

The evaluation approach that will be used for the web application will be based on Nielsen’s Heuristics and User Feedback.

The first, Nielsen’s Heuristics, will be used to evaluate the usability of the web application, while the second, User Feedback, will be gathered using a questionnaire to be filled out by a test user following usage of the system, based on the format of the original questionnaire created for this project.

These two evaluations will provide an evaluation of usability and accessibility.

**Database Evaluation**

The evaluation approach that will be used for the database will be a TPC (Transaction Processing Performance Council) styled approach using pgbench to test transactional performance under concurrent load and a CIS PostgreSQL Benchmark for testing the security of the database configuration.

These two evaluations will provide an evaluation of performance and security.

**Monitoring Platform Evaluation**

The evaluation approach that will be used for the monitoring platform will be based on the Four Golden Signals outlined by Google [128], the company which created the discipline of Site Reliability Engineering. There will be manual fault experiments performed, such as a database failure, extreme CPU usage and an application failure, whereby it is expected that the platform will provide a timely alert that staff could respond to.

This will provide an evaluation of availability.

## 5.4 Conclusions

In conclusion, the testing shall be conducted during project development, with limited aspects also to be performed by a test user towards the end of the project, mainly through the use of the web application, and the applicable testing methodology.

Evaluation, however, will only be performed at the end of the project, with the help of the same test user. The currently planned test user will be a university student who has dyslexia, which will help in assessing the accessibility of the web application. They have experience using a Virtual Learning Environment and also took part in the questionnaire. The questionnaire template will be re-used to evaluate the system, along with Nielsen’s Heuristics. If possible, more test users will be found.

With this plan in place, the non-functional requirements of availability, performance, security and accessibility will be covered and evaluated.

# 6. System Prototype

## 6.1 Introduction

The focus of this interim prototype was the implementation of the essential features, such as system access (through login), module access and assignment access. Design implementations such as accessibility have not been implemented for the Interim, as they are the finer details of a system that already has the necessary functionality.

The first step was the creation of the Virtual Machines. Inside Oracle VirtualBox, the virtual machines were setup, using the Oracle Server operating system. All three: the web application, database and monitoring platform were setup, with the last not to be developed further until after the interim.

The necessary modules and dependencies for the first and second virtual machines were installed, in order to have a working environment.

With the infrastructure setup, with all necessary frameworks and modules installed, the development of the prototype begun.

## 6.2 Prototype Development

In this section, there will be a general outline of how various parts of the system have been implemented, starting with the initial generation of code with ChatGPT and how it slots into the system, while working with various other parts.

Based on Feature Driven Development, the most essential functionality included the following four parts: the login system, dashboard, modules and assignments. These would be accessible to students and lecturers.

## 6.3 Results

In this section, there will be a usage of test cases to display the functionality that was desired for implementation in the Interim, the first four of the features lists: login, dashboard, modules and assignments. The image proof of this testing can be found in Appendix F with Test ID labelling. Only the necessary tests have been included in this section, with the remaining tests being unapplicable for the Interim.

*Web Application Testing - Student User*

|  |  |  |
| --- | --- | --- |
| Test ID | Expected Results | Actual Results |
| W-S-01 | User gains access to Dashboard. | User gains access to Dashboard. |
| W-S-02 | User gains access to Module. | User gains access to Module. |
| W-S-03 | User gains access to Assignment. | User gains access to Assignment. |
| W-S-04 | The User downloads the file or views it within their browser depending on format. | The User downloads the file or views it within their browser depending on format. |
| W-S-05 | The User downloads the file or views it within their browser depending on format. | The User downloads the file or views it within their browser depending on format. |
| W-S-06 | The user should be directed to the correct linked page. | The user should be directed to the correct linked page. |
| W-S-07 | The submission is viewable afterwards on the same page. | The submission is viewable afterwards on the same page. |
| W-S-08 | Any graded submissions should be viewable. | Any graded submissions should be viewable. |

*Web Application Testing - Lecturer User*

|  |  |  |
| --- | --- | --- |
| Test ID | Expected Results | Actual Results |
| W-L-01 | User gains access to Dashboard. | User gains access to Dashboard. |
| W-L-02 | User gains access to Module. | User gains access to Module. |
| W-L-03 | User gains access to Assignment. | User gains access to Assignment. |
| W-L-04 | Student users should be able to gain access and view updated description / attached files. | Student users should be able to gain access and view updated description / attached files. |
| W-L-05 | A new assignment should be visible within the module page. | A new assignment should be visible within the module page. |
| W-L-06 | Grade should now be available on submission. | Grade should now be available on submission. |

*Database Testing*

|  |  |  |
| --- | --- | --- |
| Test ID | Expected Results | Actual Results |
| D-A-01 | Schema should be visible in pgadmin4. | Schema should be visible in pgadmin4. |
| D-A-02 | Data should be visible in pgadmin4. | Data should be visible in pgadmin4. |

## 6.5 Conclusions

In conclusion, the system prototype is functioning as expected and provides a clear view of the feasibility of this design and implementation. This will be demonstrated as well in the demonstration in the coming weeks.

With the login system, module access, assignment / grading access and learning material access, the core functionality of the system has been implemented, allowing for an expansion of depth into these features, and the overall implementation, in the coming weeks ahead.

# 7. Issues and Future Work

## 7.1 Introduction

The purpose of this section is to discuss the issues faced during development, while outlining a plan to deal with these issues in future work.

## 7.2 Issues and Risks

The greatest issue faced by this project is the scope and required complexity. Creating a Virtual Learning Environment, even while condensed, is a difficult task, requiring a vast amount of research and design as shown throughout the document.

The first element of uncertainty, that required a different outcome than expected, in completing the interim prototype, was the deployment of a mini-orchestration technology (Minikube). It became clear that the drawbacks were numerous (issues regarding deployment with virtual machines, was the implementation valuable etc.) while it did not enhance the core point of this project, which was the application and database itself. The use of virtual machines to display the feasibility of the distributed system was sufficient, without risking unnecessary focus in the meantime that would take away from other aspects of the project.

The main uncertainties up until this point revolved around the complex requirements and research that were required to make sure this application was feasible. There was a foreboding feeling leading up to the completion of this report that there was too much to be done, and there had to be a re-evaluation.

However, with the implementation of the most complex features for the interim completed, the feasibility of the project complexity can be taken into a more positive light. The testing and evaluation later will be extensive, despite cutting down where possible, due to how much is being implemented.

## 7.3 Plans and Future Work

The priority is to implement the remaining features, which are removal features for assignments and attachments, along with the ability to create announcements globally and module locally. Furthermore, the expansion of the system to include an admin panel for the web application and the deployment of the monitoring system will make up the rest of the main functionality implementation. From there, an analysis of the user interface and a focus on accommodating non-functional requirements such as performance and accessibility will be the focus. Once these areas have been completed, having users test the application will be good to gain further feedback and find any errors that may have been missed. Once that is completed, an evaluation will take place to determine what needs to be done to complete the project.

### 7.3.1 Project Plan with GANTT Chart

In order to schedule this, I have created a GANTT Chart with a realistic timeline to implement the outlined plans to complete the project, accounting for the exam period and assignments that will be given in the second semester, where possible.

# 8. Appendices

## Appendix A: Potential Technologies Research

These technologies, while not necessarily core to the project, have been briefly researched as potential options to be implemented after the completion of the interim prototype. They are focused on the deployment of the application for use by the tens of thousands of potential students, and the relevant lecturers. While deeper research has not been conducted on these technologies just yet, with an exclusion of the advantages and disadvantages of each, due to no planned initial (or potentially any) implementation, they will be kept in mind as the project evolves.

**Caching**

When the same data is frequently pulled from a database, it begins to drag on the system, when you do not use caching. While some database technologies, such as PostgreSQL, provide caching through functionality such as a shared buffer cache, this is not the case for all. Caching is used to store copies of frequently accessed data, temporarily, in a high-speed location, in order to limit the strain on the database. While PostgreSQL will be used, which does provide its own solution for caching, in the future, for the sake of scalability, there will be a consideration for the implementation of a caching software solution, to help the system cope with the data load.

***Redis***

Created as a data store in-memory caching technology, it is designed for extreme speed and low latency. It supports rich data-structures which can be accessed by commands and is typically used to cache session storage and database results (from queries etc.). It is often used in modern distributed systems [129].

***Valkey***

Created as an alternative open-source solution to Redis, by a number of former Redis developers, in response to changes to the licensing of Redis, it focuses on core design principles such as stability and observability, providing much of the same features as Redis itself [130].

***Dragonfly***

Designed to be efficient caching technology for heavy workloads, focusing on stable performance, this software solution is designed for modern hardware in non-clustered configurations (which would be potentially an issue in implementing this for Eagna) [131].

**Container Orchestration Technologies**

While Containers are packages of created software, such as a web application, joined together with all the relevant technologies required to run, these Container Orchestration Technologies focus on the deployment and co-ordination of these containers across a cluster of machines, designed to help availability, scalability and reliability, by scheduling and releasing the release of updates, and the redeployment of containers, in the case of container failure.

Having originally considered running the system on a localized container orchestration package such as Minikube or k3s, it does not add anything to the actual prototype of the system, only for the feasibility of deployment at a larger scale. Because of this, the focus has instead switched to the intricacies of the system for the moment, with the consideration in design for potential deployment later on, with one of these technologies.

***Docker***

This is a very popular container orchestration technology focused on providing a lightweight solution and has been used in the TU856 course for multiple modules, such as Cloud Computing and Advanced Databases, for having an isolated environment for test development. It is typically used for microservices, CI / CD pipelines and development environments [132].

***Kubernetes***

This is an open-source container orchestration platform designed to automate deployment, scalability and management of multiple nodes of a distributed system, on the cloud or in on-site hardware. It uses pods (groups of these containers) deployed on cluster nodes (which in essence gives you copies of your entire system) to ensure availability and up-time of the service you are providing [133].

***ECS***

An Amazon product, this is a container orchestration technology that is actually designed to be serverless (avoiding cluster nodes like in Kubernetes). It does not provide the same level of in-depth control over your containers and is intended to be a more ‘lightweight’ solution for container orchestration [134]. Amazon provides the option to use Fargate to host ECS, where you do not need to manage the server, only focusing on application development [135].

**Cloud Services**

Ran primarily by large corporations, Cloud Services provide rentable servers, storage, platforms and databases, that are accessible through the Internet. In the past, where systems would be deployed on local, on-site hardware (servers etc.), the advent of Cloud Services now allows a customer to rely on off-site rentable resources, where they can deploy what systems they want, without concerning themselves with the day-to-day maintenance of the hardware and infrastructure. In regard to this project, if there was a desire to deploy this system off-site, wanting to avoid the worry of hosting infrastructure on-site, while keeping the homegrown solution of Eagna, these services would provide the answer, depending on their price.

***Amazon AWS***

Provided by Amazon, the AWS platform provides a variety on-demand computing resources, storage, data storage, databases, networking, security, analytics, and tools for maintenance [136]. The global availability of AWS is not to be understated, using over ‘nine million kilometres of fibre optic cabling’ to support its network at rapid and reliable speeds. It has 120 availability zones (locations to physically support your rented service), including three in Ireland (which form eu-west-1) [137].

***Microsoft Azure***

Provided by Microsoft, the Azure platform provides on-demand computing resources, storage, data storage, databases and networking [138]. It has over 70 availability zones (regions) in comparison to AWS, but it also provides three in Ireland [139].

***Google Cloud***

Provided by Google, the Cloud platform provides virtual machines, a Kubernetes implementation (Google Kubernetes Engine), serverless containers, private networks, data storage, databases and analytics [140]. It has 127 availability zones, the greatest amount among the three solutions, however, it does not host directly in Ireland (other EU nations are available however, to help with easier compliance in GDPR, even if not directly in Ireland).

**Potential Technologies Conclusion**

Out of the potential technologies, for a theoretical real-time deployment, the most relevant technologies briefly researched and likely to be research further would be Redis, Kubernetes and Amazon AWS.

There are financial limitations for deployment on the Cloud Services, which make it unlikely to be implemented for the final project.

## Appendix B: Questionnaire Feedback

**Student Feedback**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 58 - Student Feedback Appendix - Part I

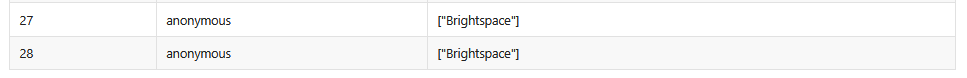


Figure 59 - Student Feedback Appendix - Part II

A screenshot of a computer

AI-generated content may be incorrect.

Figure 60 - Student Feedback Appendix - Part III

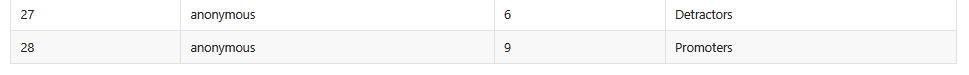


Figure 61 - Student Feedback Appendix - Part IV

A screenshot of a computer

AI-generated content may be incorrect.

Figure 62 - Student Feedback Appendix - Part V

A screenshot of a phone

AI-generated content may be incorrect.

Figure 63 - Student Feedback Appendix - Part VI

A screenshot of a computer

AI-generated content may be incorrect.

Figure 64 - Student Feedback Appendix - Part VII

A screenshot of a computer

AI-generated content may be incorrect.

Figure 65 - Student Feedback Appendix - Part VIII

A screenshot of a computer

AI-generated content may be incorrect.

Figure 66 - Student Feedback Appendix - Part IX

A screenshot of a chat

AI-generated content may be incorrect.

Figure 67 - Student Feedback Appendix - Part X

A screenshot of a computer

AI-generated content may be incorrect.

Figure 68 - Student Feedback Appendix - Part XI



Figure 69 - Student Feedback Appendix - Part XII

A screenshot of a computer

AI-generated content may be incorrect.

Figure 70 - Student Feedback Appendix - Part XIII

A blue circle with black text

AI-generated content may be incorrect.

Figure 71 - Student Feedback Appendix - Part XIV

**Lecturer Feedback**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 72 - Lecturer Feedback Appendix - Part I

A screenshot of a computer

AI-generated content may be incorrect.

Figure 73 - Lecturer Feedback Appendix - Part II

A screenshot of a computer

AI-generated content may be incorrect.

Figure 74 - Lecturer Feedback Appendix - Part III

A screenshot of a computer

AI-generated content may be incorrect.

Figure 75 - Lecturer Feedback Appendix - Part IV

A screenshot of a computer

AI-generated content may be incorrect.

Figure 76 - Lecturer Feedback Appendix - Part V

A screenshot of a computer

AI-generated content may be incorrect.

Figure 77 - Lecturer Feedback Appendix - Part VI

A screenshot of a computer

AI-generated content may be incorrect.

Figure 78 - Lecturer Feedback Appendix - Part VII

A screenshot of a computer

AI-generated content may be incorrect.

Figure 79 - Lecturer Feedback Appendix - Part VIII

A blue circle with black text

AI-generated content may be incorrect.

Figure 80 - Lecturer Feedback Appendix - Part IX

## Appendix C: System Model and Analysis

A diagram of a business process

AI-generated content may be incorrect.

Figure 81 - Initial Student Use Case (Blown Up)

A diagram of a software system

AI-generated content may be incorrect.

Figure 82 - Initial Lecturer Use Case (Blown Up)

A diagram of a company

AI-generated content may be incorrect.

Figure 83 - Expanded Use Case (Blown Up)

## Appendix D: Design

A diagram of a software development network

AI-generated content may be incorrect.

Figure 84 - Eagna System Deployment (Blown Up)

A screenshot of a login screen

AI-generated content may be incorrect.

Figure 85 - Wireframes - Login System – Blown Up

A screenshot of a computer dashboard

AI-generated content may be incorrect.

Figure 86 - Wireframes - Student Dashboard – Blown Up

A screenshot of a computer dashboard

AI-generated content may be incorrect.

Figure 87 - Wireframes - Lecturer Dashboard – Blown Up

A blue and white web page

AI-generated content may be incorrect.

Figure 88 - Wireframes - Student Modules – Blown Up

A blue and white web page

AI-generated content may be incorrect.

Figure 89 - Wireframes - Lecturer Modules – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 90 - Wireframes - Student Assignments – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 91 - Wireframes - Lecturer Assignments – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 92 - Wireframes - Create Assignment – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 93 - Wireframes - Add / Edit Assignment Grade – Blown Up

A screenshot of a login form

AI-generated content may be incorrect.

Figure 94 - Wireframes - Student Registration – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 95 - Wireframes - Student User Profile – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 96 - Wireframes - Lecturer User Profile – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 97 - Wireframes - Create Announcements (Same Template for Global / Module) – Blown Up

A screenshot of a computer

AI-generated content may be incorrect.

Figure 98 - External Tools Page (Student & Lecturer) – Blown Up

A blue and white box with text

AI-generated content may be incorrect.

Figure 99 - Wireframes - Administrator Panel – Blown Up

A blue and white login screen

AI-generated content may be incorrect.

Figure 100 - Wireframes - Add Module (Admin) – Blown Up

A blue and white login screen

AI-generated content may be incorrect.

Figure 101 - Wireframes - Add Lecturer (Admin) – Blown Up

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 102 - Wireframes - Edit Enrolment (Admin) – Blown Up



Figure 103 - Entity Relationship Diagram (Blown Up)

A diagram of a computer

AI-generated content may be incorrect.

## Appendix E: Prompts Used with ChatGPT

1. I have outlined a rough idea of my schema that would work with Django, HTMX and PostgreSQL for my VLE, are there any issues you see looking at this? (Attached outline.txt)
2. I want to focus on creating a working prototype, so let’s break it down into individual pieces that I can implement into my working environment and edit to my needs as we go. Starting off, let us create the login system (for students and lecturers only), so that they can gain access to their homepage / dashboard. I will need all relevant files for construction, html, css, js (where needed), htmx, django. Also, give me assistance on initial directory placement.
3. Ok, I want to begin building the dashboard code. There will be slight differences between what is presented for students and lecturers, but I will deliver the outline. Student Dashboard: - Header (Full Width) -- Left: 'Eagna' Text -- Center: Nav Menu: Dashboard | Tools | Inbox | Website -- Right: Notification Button | User (Username) - Body (Full Width with Small Padding Around Elements -- Upcoming Assignments (Show Title) --- Clickable Box Item for Each Assignment Upcoming (Title, Module, Due Date displayed) --> will display a modal displaying relevant information -- Your Modules (Show Title) --- Clickable Box Item for Each Module (Title, Module Code, Lecturer displayed) --> when click, will bring to module page - Footer (Full Width) -- Image --- Same Nav Menu As Header Lecturer Dashboard Differences: - Instead of Upcoming Assignments -- Clickable Box Item for each Ungraded Submissions (Title, Module, Student) --> will display a modal displaying relevant information
4. Now, I want to start working on the weekly resources in the module page. I want the sections to be split between weeks (1 - 15) by default, and the weeks should only appear when there has been a learning material file attached to them. The main title will be 'Academic Calendar', with a below subheading with the Week Number, a below small description editable by the lecturer, and then each attached file to that week below. There should be a usable upload button for the lecturer to attach files.
5. the next step is to add descriptions for assignments, before adding the ability for lecturers to create assignments (and attaching files to it that the student can view)
6. I want students to be able to make submissions to assignments now, and for lecturers to be able to grade these submissions.
7. I need the dashboard for lecturers / students for upcoming assignments and ungraded submissions to be clicked on and display the assignment page we have implemented. There is no need for the modals we created in step 3 anymore.

## Appendix F: Testing Results

*Web Application Testing - Student User*

W-S-01  
W-S-02  
W-S-03  
W-S-04  
W-S-05  
W-S-06  
W-S-07  
W-S-08

*Web Application Testing - Lecturer User*

W-L-01  
W-L-02  
W-L-03  
W-L-04  
W-L-05  
W-L-06

*Database Testing*

D-A-01  
D-A-02

## Appendix G: Additional Code Samples

# 9. References

[1] J. Nielsen, ‘Jakob’s Law of the Internet User Experience’, UX Tigers. Accessed: Oct. 29, 2025. [Online]. Available: https://www.uxtigers.com/post/jakobs-law

[2] ‘End of Web Design’, Nielsen Norman Group. Accessed: Oct. 29, 2025. [Online]. Available: https://www.nngroup.com/articles/end-of-web-design/

[3] T. U. Dublin, ‘Connect | VLE - Brightspace | TU Dublin’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.tudublin.ie/connect/vle/

[4] ‘The Current and Future State of LMSs’. Accessed: Nov. 05, 2025. [Online]. Available: https://ikee.lib.auth.gr/record/306354/files/GRI-2019-25105.pdf

[5] ‘Adopting D2L Brightspace for a consistent user experience’, D2L. Accessed: Nov. 05, 2025. [Online]. Available: https://www.d2l.com/en-mea/why-d2l/customers/tu-dublin/

[6] ‘Changes to the Virtual Learning Environment Standards - UCD IT Services’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.ucd.ie/itservices/ourservices/educationaltechnologies/digitallearning-instructors/changestothevirtuallearningenvironmentstandards/

[7] ‘Standard Operation Procedures for the VLE Structure - University of Limerick’. Accessed: Nov. 05, 2025. [Online]. Available: https://www.ul.ie/media/58117/download?inline

[8] ‘Canvas | University of Oxford’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.ox.ac.uk/students/academic/guidance/canvas

[9] ‘UCD Connect’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.ucd.ie/connect/

[10] ‘About Loop | DCU’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.dcu.ie/teu/loop

[11] ‘Moodle | DCU’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.dcu.ie/students/moodle

[12] ‘Moodle | Maynooth University’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.maynoothuniversity.ie/centre-teaching-and-learning/technology-enhanced-learning/moodle

[13] T. C. Dublin, ‘Blackboard Learn VLE - IT Services | Trinity College Dublin’. Accessed: Oct. 12, 2025. [Online]. Available: https://www.tcd.ie/itservices/our-services/blackboard-learn-vle/

[14] ‘Login - Elearning - Western Michigan University’. Accessed: Nov. 05, 2025. [Online]. Available: https://elearning.wmich.edu/d2l/login?noRedirect=1

[15] ‘Welcome to the Higher Education Instructor Knowledge Base’, Brightspace. Accessed: Nov. 05, 2025. [Online]. Available: https://community.d2l.com/brightspace/kb/articles/25590-welcome-to-the-higher-education-instructor-knowledge-base

[16] ‘Create discussion forums and topics’, Brightspace. Accessed: Nov. 05, 2025. [Online]. Available: https://community.d2l.com/brightspace/kb/articles/3454-create-discussion-forums-and-topics

[17] ‘About Class Progress’, Brightspace. Accessed: Nov. 05, 2025. [Online]. Available: https://community.d2l.com/brightspace/kb/articles/3314-about-class-progress

[18] ‘About Portfolio’, Brightspace. Accessed: Nov. 05, 2025. [Online]. Available: https://community.d2l.com/brightspace/kb/articles/5477-about-portfolio

[19] ‘Resources’, Brightspace. Accessed: Nov. 05, 2025. [Online]. Available: https://community.d2l.com/brightspace/categories/resources

[20] bentleyc, ‘Adding and Using Forums in Moodle’, Technology Help. Accessed: Nov. 05, 2025. [Online]. Available: https://help.lafayette.edu/adding-and-using-forums-in-moodle/

[21] ‘Features - MoodleDocs’. Accessed: Nov. 05, 2025. [Online]. Available: https://docs.moodle.org/501/en/Features

[22] ‘Moodle Plugins directory | Moodle.org’. Accessed: Nov. 05, 2025. [Online]. Available: https://moodle.org/plugins/index.php

[23] ‘Technology solutions for the global education community | Turnitin’. Accessed: Nov. 05, 2025. [Online]. Available: https://www.turnitin.com/

[24] T. C. Dublin, ‘Panopto Lecture Capture - IT Services | Trinity College Dublin’. Accessed: Nov. 05, 2025. [Online]. Available: https://www.tcd.ie/itservices/our-services/panopto-lecture-capture/

[25] ‘Class Collaborate (formerly Blackboard Collaborate)’, Class. Accessed: Nov. 05, 2025. [Online]. Available: https://www.class.com/collaborate/

[26] ‘Why Choose Blackboard’, Anthology. Accessed: Nov. 05, 2025. [Online]. Available: https://www.anthology.com/why-choose-blackboard

[27] ‘AI Design Assistant’. Accessed: Nov. 05, 2025. [Online]. Available: https://help.blackboard.com/Learn/Instructor/Ultra/Course\_Content/Create\_Content/AI\_Design\_Assistant

[28] ‘Content Designer: Enhanced Documents – 3900.98 | Blackboard Help’. Accessed: Nov. 05, 2025. [Online]. Available: https://help.blackboard.com/node/48536

[29] ‘Anthology Digital Assistant’, Anthology. Accessed: Nov. 05, 2025. [Online]. Available: https://www.anthology.com/products/lifecycle-engagement/enrollment-and-retention/anthology-digital-assistance

[30] ‘Students - University of Galway’. Accessed: Nov. 05, 2025. [Online]. Available: https://www.universityofgalway.ie/information-solutions-services/services-for-students/canvas/students/

[31] ‘Canvas’, Queen’s DigiHub. Accessed: Nov. 05, 2025. [Online]. Available: https://blogs.qub.ac.uk/digitallearning/staff/digital-teaching/qub-digital-tools/canvas/

[32] ‘Canvas by Instructure: World Leading LMS for Teaching & Learning’, Instructure. Accessed: Nov. 05, 2025. [Online]. Available: https://www.instructure.com/canvas

[33] ‘Loop | Teaching Enhancement Unit’. Accessed: Nov. 05, 2025. [Online]. Available: https://www.dcu.ie/teu/loop-teaching-enhancement-unit

[34] ‘One platform to connect’, Zoom. Accessed: Nov. 05, 2025. [Online]. Available: https://www.zoom.com

[35] ‘Ouriginal: Text-matching solution - Plagiarism prevention’. Accessed: Nov. 05, 2025. [Online]. Available: https://ouriginal.com/

[36] ‘Home - Mahara ePortfolio System’. Accessed: Nov. 05, 2025. [Online]. Available: https://mahara.org/

[37] ‘D2L Pricing’. Accessed: Nov. 06, 2025. [Online]. Available: https://assets.applytosupply.digitalmarketplace.service.gov.uk/g-cloud-14/documents/93450/655121115446377-pricing-document-2024-05-06-1950.pdf

[38] B. Warwick, ‘Blackboard Pricing’.

[39] ‘Canvas Pricing’. Accessed: Nov. 06, 2025. [Online]. Available: https://assets.applytosupply.digitalmarketplace.service.gov.uk/g-cloud-14/documents/709019/173463782527186-pricing-document-2024-05-01-1108.pdf

[40] ‘Moodle Pricing’, MoodleCloud. Accessed: Nov. 06, 2025. [Online]. Available: https://www.moodlecloud.com/pricing/

[41] T. U. Dublin, ‘Welcome to TU Dublin | Student Life | TU Dublin’. Accessed: Nov. 06, 2025. [Online]. Available: https://www.tudublin.ie/for-students/starting-at-tu-dublin/student-life/

[42] R. Larsen, *Beginning HTML and CSS*. John Wiley & Sons, 2013.

[43] ‘HTML Standard’. Accessed: Nov. 05, 2025. [Online]. Available: https://html.spec.whatwg.org/multipage/

[44] ‘Usage Statistics and Market Share of HTML for Websites, November 2025’. Accessed: Nov. 06, 2025. [Online]. Available: https://w3techs.com/technologies/details/ml-html\_any

[45] ‘Cascading Style Sheets’. Accessed: Nov. 06, 2025. [Online]. Available: https://www.w3.org/Style/CSS/Overview.en.html

[46] ‘Usage Statistics of CSS for Websites, November 2025’. Accessed: Nov. 06, 2025. [Online]. Available: https://w3techs.com/technologies/details/ce-css

[47] ‘Usage Statistics of JavaScript as Client-side Programming Language on Websites, November 2025’. Accessed: Nov. 06, 2025. [Online]. Available: https://w3techs.com/technologies/details/cp-javascript

[48] ‘Usage Statistics and Market Share of JavaScript Libraries for Websites, November 2025’. Accessed: Nov. 06, 2025. [Online]. Available: https://w3techs.com/technologies/overview/javascript\_library

[49] ‘</> htmx ~ Documentation’. Accessed: Nov. 06, 2025. [Online]. Available: https://htmx.org/docs/

[50] ‘Creating a React App – React’. Accessed: Nov. 06, 2025. [Online]. Available: https://react.dev/learn/creating-a-react-app

[51] ‘Quick Start – React’. Accessed: Nov. 06, 2025. [Online]. Available: https://react.dev/learn

[52] ‘React Reference Overview – React’. Accessed: Nov. 06, 2025. [Online]. Available: https://react.dev/reference/react

[53] V. Gowda and S. Rangaswamy, ‘Addressing the Limitations of React JS’, vol. 07, no. 04, 2020.

[54] ‘Django overview’, Django Project. Accessed: Nov. 06, 2025. [Online]. Available: https://www.djangoproject.com/start/overview/

[55] ‘Django 2.0 release notes | Django documentation’, Django Project. Accessed: Nov. 06, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/releases/2.0/

[56] ‘User authentication in Django | Django documentation’, Django Project. Accessed: Nov. 06, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/auth/

[57] S. Dhadil *et al.*, ‘Django Unleashed: A Deep Dive into the Features and Advantages of the Django Framework’, *rjps*, vol. 14, no. 3, 2024, doi: 10.26463/rjps.14\_3\_7.

[58] ‘Design Decisions in Flask — Flask Documentation (3.1.x)’. Accessed: Nov. 06, 2025. [Online]. Available: https://flask.palletsprojects.com/en/stable/design/

[59] D. Ghimire, ‘Comparative study on Python web frameworks: Flask and Django’.

[60] ‘Rails Routing from the Outside In’, Ruby on Rails Guides. Accessed: Nov. 06, 2025. [Online]. Available: https://guides.rubyonrails.org/routing.html

[61] ‘Active Record Basics’, Ruby on Rails Guides. Accessed: Nov. 06, 2025. [Online]. Available: https://guides.rubyonrails.org/active\_record\_basics.html

[62] ‘Securing Rails Applications’, Ruby on Rails Guides. Accessed: Nov. 06, 2025. [Online]. Available: https://guides.rubyonrails.org/security.html

[63] C. Scott and B. Lewis, ‘Advancements in Web Application Development: An Analytical Review of Ruby on Rails, Python Frameworks, and Cloud-Centric Solutions’, Accessed: Nov. 07, 2025. [Online]. Available: https://www.researchgate.net/profile/Olatunji-Isreal/publication/394759430\_Advancements\_in\_Web\_Application\_Development\_An\_Analytical\_Review\_of\_Ruby\_on\_Rails\_Python\_Frameworks\_and\_Cloud-Centric\_Solutions/links/68a5f7836327cf7b63d842b4/Advancements-in-Web-Application-Development-An-Analytical-Review-of-Ruby-on-Rails-Python-Frameworks-and-Cloud-Centric-Solutions.pdf

[64] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, ‘SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review’, *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 97, June 2023, doi: 10.3390/bdcc7020097.

[65] P. G. D. Group, ‘PostgreSQL’, PostgreSQL. Accessed: Nov. 07, 2025. [Online]. Available: https://www.postgresql.org/

[66] ‘PostgreSQL: About’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.postgresql.org/about/

[67] S. V. Salunke and A. Ouda, ‘A Performance Benchmark for the PostgreSQL and MySQL Databases’, *Future Internet*, vol. 16, no. 10, p. 382, Oct. 2024, doi: 10.3390/fi16100382.

[68] ‘MySQL’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.mysql.com/

[69] ‘MySQL :: MySQL Documentation’. Accessed: Nov. 07, 2025. [Online]. Available: https://dev.mysql.com/doc/

[70] ‘MongoDB: The World’s Leading Modern Database’, MongoDB. Accessed: Nov. 07, 2025. [Online]. Available: https://www.mongodb.com/

[71] ‘Apache Cassandra | Apache Cassandra Documentation’, Apache Cassandra. Accessed: Nov. 07, 2025. [Online]. Available: https://cassandra.apache.org/

[72] A. Chebotko, A. Kashlev, and S. Lu, ‘A Big Data Modeling Methodology for Apache Cassandra’, in *2015 IEEE International Congress on Big Data*, June 2015, pp. 238–245. doi: 10.1109/BigDataCongress.2015.41.

[73] ‘Home’, KDM. Accessed: Nov. 07, 2025. [Online]. Available: https://kdm89225124.wordpress.com/home/

[74] ‘Prometheus - Monitoring system & time series database’. Accessed: Nov. 07, 2025. [Online]. Available: https://prometheus.io/

[75] M. Holopainen, ‘Monitoring Container Environment with Prometheus and Grafana’.

[76] ‘Grafana: The open and composable observability platform’, Grafana Labs. Accessed: Nov. 07, 2025. [Online]. Available: https://grafana.com/

[77] ‘Grafana | Query, visualize, alerting observability platform’, Grafana Labs. Accessed: Nov. 07, 2025. [Online]. Available: https://grafana.com/grafana/

[78] ‘Zabbix: The enterprise-class open source observability solution’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.zabbix.com/

[79] P. Uytterhoeven and R. Olups, *Zabbix 4 Network Monitoring: Monitor the performance of your network devices and applications using the all-new Zabbix 4.0, 3rd Edition*. Packt Publishing Ltd, 2019.

[80] Datadog, ‘Cloud Monitoring as a Service’, Datadog. Accessed: Nov. 07, 2025. [Online]. Available: https://www.datadoghq.com/

[81] Datadog, ‘Infrastructure & Application Monitoring as a Service’, Datadog. Accessed: Nov. 07, 2025. [Online]. Available: https://www.datadoghq.com/product/

[82] ‘Pricing | Datadog’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.datadoghq.com/pricing/

[83] D. T. Vojnak, B. S. Ðor\djević, V. V. Timčenko, and S. M. Štrbac, ‘Performance Comparison of the type-2 hypervisor VirtualBox and VMWare Workstation’, in *2019 27th Telecommunications Forum (℡FOR)*, IEEE, 2019, pp. 1–4. Accessed: Nov. 07, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8971213/

[84] ‘Oracle VirtualBox’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.virtualbox.org/

[85] B. Djordjević, V. Timčenko, N. Kraljević, I. Jovičić, and N. Davidović, ‘Comparison of VMware Workstation, VirtualBox and MS Hyper-V hypervisor performance with MS Windows OS based guests’, in *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, Mar. 2023, pp. 1–5. doi: 10.1109/INFOTEH57020.2023.10094080.

[86] ‘VMware by Broadcom - Cloud Computing for the Enterprise’. Accessed: Nov. 07, 2025. [Online]. Available: https://www.vmware.com

[87] ‘</> htmx ~ A Real World React -> htmx Port’. Accessed: Nov. 05, 2025. [Online]. Available: https://htmx.org/essays/a-real-world-react-to-htmx-port/

[88] ‘django-htmx 1.26.0 documentation’. Accessed: Nov. 07, 2025. [Online]. Available: https://django-htmx.readthedocs.io/en/latest/index.html

[89] A. Carzaniga, A. Mattavelli, and M. Pezzè, ‘Measuring Software Redundancy’, in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, May 2015, pp. 156–166. doi: 10.1109/ICSE.2015.37.

[90] C. Majors, L. Fong-Jones, and G. Miranda, *Observability Engineering*. O’Reilly Media, Inc., 2022.

[91] J. Garcia, I. Ivkovic, and N. Medvidovic, ‘A comparative analysis of software architecture recovery techniques’, in *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Nov. 2013, pp. 486–496. doi: 10.1109/ASE.2013.6693106.

[92] D. Wessels, *Web Caching*. O’Reilly Media, Inc., 2001.

[93] ‘Django’s cache framework | Django documentation’, Django Project. Accessed: Nov. 18, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/cache/

[94] E. Inersjö, ‘Comparing database optimisation techniques in PostgreSQL’.

[95] D. Byrne, *Full Stack Python Security: Cryptography, TLS, and attack resistance*. Simon and Schuster, 2021.

[96] S. Chen, S. Ahmmed, K. Lal, and C. Deming, ‘Django Web Development Framework: Powering the Modern Web’, *Am. j. trade policy*, vol. 7, no. 3, pp. 99–106, Dec. 2020, doi: 10.18034/ajtp.v7i3.675.

[97] ‘General Data Protection Regulation (GDPR) – Legal Text’, General Data Protection Regulation (GDPR). Accessed: Nov. 13, 2025. [Online]. Available: https://gdpr-info.eu/

[98] ‘Art. 5 GDPR – Principles relating to processing of personal data’, General Data Protection Regulation (GDPR). Accessed: Nov. 13, 2025. [Online]. Available: https://gdpr-info.eu/art-5-gdpr/

[99] ‘Art. 33 GDPR – Notification of a personal data breach to the supervisory authority’, General Data Protection Regulation (GDPR). Accessed: Nov. 13, 2025. [Online]. Available: https://gdpr-info.eu/art-33-gdpr/

[100] ‘EU-US data transfers - European Commission’. Accessed: Nov. 13, 2025. [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/international-dimension-data-protection/eu-us-data-transfers\_en

[101] ‘Art. 35 GDPR – Data protection impact assessment’, General Data Protection Regulation (GDPR). Accessed: Nov. 13, 2025. [Online]. Available: https://gdpr-info.eu/art-35-gdpr/

[102] ‘Art. 37 GDPR – Designation of the data protection officer’, General Data Protection Regulation (GDPR). Accessed: Nov. 13, 2025. [Online]. Available: https://gdpr-info.eu/art-37-gdpr/

[103] ‘Vision impairment and blindness’. Accessed: Nov. 13, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[104] ‘Deafness and hearing loss’. Accessed: Nov. 13, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[105] T. U. Dublin, ‘Student Wellbeing | Disability Support Service | TU Dublin’. Accessed: Nov. 13, 2025. [Online]. Available: https://www.tudublin.ie/for-students/student-services-and-support/student-wellbeing/disability-support-service/

[106] ‘Speech Sound Impairment.pdf’. Accessed: Nov. 13, 2025. [Online]. Available: https://enableireland.ie/sites/default/files/publication/Speech%20Sound%20Impairment.pdf

[107] ‘Learning Disability-R4 final NE.pdf’. Accessed: Nov. 13, 2025. [Online]. Available: https://enableireland.ie/sites/default/files/publication/Learning%20Disability-R4%20final%20NE.pdf

[108] J. den Houting, ‘Neurodiversity: An insider’s perspective’, *Autism*, vol. 23, no. 2, pp. 271–273, Feb. 2019, doi: 10.1177/1362361318820762.

[109] ‘Noncommunicable diseases’. Accessed: Nov. 13, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

[110] ‘Mental disorders’. Accessed: Nov. 13, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/mental-disorders

[111] ‘Disability in Ireland - Factsheet 2024’, Disability Federation of Ireland. Accessed: Nov. 13, 2025. [Online]. Available: https://www.disability-federation.ie/publications/disability-in-ireland-factsheet-2024/

[112] B. McCormack and D. Lawless, ‘Suaimhneas: A responsive web application for anxiety management and wellbeing Final Project Report’, TU Dublin, Dublin, 2023.

[113] D. Hogan and P. Bourke, ‘An eLearning system for anonymous feedback session with built-in polling using node.js’, DIT, Dublin, 2014.

[114] J. Nielsen, ‘Enhancing the Explanatory Power of Usability Heuristics’.

[115] J. Nielsen, ‘How I Developed the 10 Usability Heuristics’, UX Tigers. Accessed: Nov. 13, 2025. [Online]. Available: https://www.uxtigers.com/post/usability-heuristics-history

[116] ‘Database access optimization | Django documentation’, Django Project. Accessed: Nov. 18, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/db/optimization/

[117] ‘https://ukhomeoffice.github.io/accessibility-posters/posters/accessibility-posters.pdf’. Accessed: Nov. 18, 2025. [Online]. Available: https://ukhomeoffice.github.io/accessibility-posters/posters/accessibility-posters.pdf

[118] ‘Why (and How) You Should Use Feature-Driven Development’, Lucidchart. Accessed: Nov. 13, 2025. [Online]. Available: https://www.lucidchart.com/blog/why-use-feature-driven-development

[119] ‘What is FDD in Agile?’, Planview. Accessed: Nov. 13, 2025. [Online]. Available: https://www.planview.com/resources/articles/fdd-agile/

[120] A. Firdaus, I. Ghani, and N. I. M. Yasin, ‘Developing secure websites using feature driven development (FDD): a case study’, *Journal of Clean Energy Technologies*, vol. 1, no. 4, pp. 322–326, 2013.

[121] S. Romano, D. Fucci, G. Scanniello, B. Turhan, and N. Juristo, ‘Findings from a multi-method study on test-driven development’, *Information and Software Technology*, vol. 89, pp. 64–77, 2017.

[122] H. Berger and P. Beynon‐Davies, ‘The utility of rapid application development in large‐scale, complex projects’, *Information Systems Journal*, vol. 19, no. 6, pp. 549–570, Nov. 2009, doi: 10.1111/j.1365-2575.2009.00329.x.

[123] ‘Accessible colour palettes’. Accessed: Nov. 20, 2025. [Online]. Available: https://data.europa.eu/apps/data-visualisation-guide/accessible-colour-palettes?utm\_source=chatgpt.com

[124] ‘Using the Django authentication system | Django documentation’, Django Project. Accessed: Nov. 20, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/auth/default/

[125] ‘View decorators | Django documentation’, Django Project. Accessed: Nov. 20, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/http/decorators/

[126] ‘django.contrib.auth | Django documentation’, Django Project. Accessed: Nov. 20, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/ref/contrib/auth/

[127] ‘Django settings | Django documentation’, Django Project. Accessed: Nov. 20, 2025. [Online]. Available: https://docs.djangoproject.com/en/5.2/topics/settings/

[128] ‘Google SRE monitoring ditributed system - sre golden signals’. Accessed: Nov. 22, 2025. [Online]. Available: https://sre.google/sre-book/monitoring-distributed-systems/#xref\_monitoring\_golden-signals

[129] Redis, ‘Redis - The Real-time Data Platform’, Redis. Accessed: Nov. 08, 2025. [Online]. Available: https://redis.io/

[130] ‘Valkey’. Accessed: Nov. 08, 2025. [Online]. Available: https://valkey.io/

[131] ‘Dragonfly | An In-Memory Data Store without Limits’, Dragonfly. Accessed: Nov. 08, 2025. [Online]. Available: https://www.dragonflydb.io

[132] ‘Docker: Accelerated Container Application Development’. Accessed: Nov. 08, 2025. [Online]. Available: https://www.docker.com/

[133] ‘Production-Grade Container Orchestration’, Kubernetes. Accessed: Nov. 08, 2025. [Online]. Available: https://kubernetes.io/

[134] ‘Amazon ECS’, Amazon Web Services, Inc. Accessed: Nov. 08, 2025. [Online]. Available: https://aws.amazon.com/ecs/

[135] ‘Serverless Compute - AWS Fargate - AWS’, Amazon Web Services, Inc. Accessed: Nov. 08, 2025. [Online]. Available: https://aws.amazon.com/fargate/

[136] ‘Cloud Products’, Amazon Web Services, Inc. Accessed: Nov. 08, 2025. [Online]. Available: https://aws.amazon.com/products/

[137] ‘Global Infrastructure’, Amazon Web Services, Inc. Accessed: Nov. 08, 2025. [Online]. Available: https://aws.amazon.com/about-aws/global-infrastructure/

[138] ‘Cloud Computing Services | Microsoft Azure’. Accessed: Nov. 08, 2025. [Online]. Available: https://azure.microsoft.com/en-us

[139] ‘Explore | Azure global infrastructure experience’. Accessed: Nov. 08, 2025. [Online]. Available: https://datacenters.microsoft.com/globe/explore?view=table

[140] ‘Free Trial and Free Tier Services and Products’, Google Cloud. Accessed: Nov. 08, 2025. [Online]. Available: https://cloud.google.com/free