



# **Final Year Project Proposal**

**TU856**

**Eagna – An Integrated VLE for TUD**

**Conor Davis**

**C20441826**

School of Computer Science  
TU Dublin – City Campus

**24 / 10 / 2025**

---

Table of Contents

Contents

**Table of Contents ..... i**

**Declaration ..... ii**

***Summary..... 1***

***Background (and References) ..... 2***

***Proposed Approach ..... 3***

***Deliverables ..... 5***

***Project Schedule ..... 5***

***Technical Requirements..... 6***

***Conclusion ..... 6***

***References..... 1***

***Appendix A: First Project Review..... 1***

***Appendix B: Second Project Review ..... 1***

***Appendix C: Prompts Used with ChatGPT ..... 1***

***Appendix D:..... 1***

## 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

24/10/2025

## *Summary*

My project idea, Eagna (the word 'wisdom' in Irish), 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, into a single online portal, combining the functionality of a Virtual Learning Environment with an integrated registration system, providing access to the necessary tools and functionality needed to engage in our studies.

The core design of this system would be a web application serving as the central learning hub for the university: handling student registration, including their assignment to mandatory or optional modules, where they will then be able to engage in learning, through module access to notes, videos, assignments and grades. Lecturers will also have the means to access their specific modules, to add their notes and videos, the ability to upload and set assignments, delivering grades and communicating with their students. To support this web application, I will create a database alongside a system monitoring platform, to manage reliability and availability, forming my distributed system.

With this project, the system can be personalised and tailored towards the unique needs of the college, including the security, accessibility and availability concerns associated with such a system.

The use of separate software services, that have to be rented, organized and then connected together, is diminished. The recent issues regarding the new registration system, which then affected Brightspace, inspired the idea of this platform. Like the various institutions that also use their own hub to link everything together, this application would deliver an insight into the feasibility of a custom platform for TUD.

On a personal level, this project will give me an insight on all the facets of creating a full-stack system, from user interface design and accessibility, system design, site reliability engineering, database management and more. When I leave TUD, I hope to become a Site Reliability Engineer or a Full-Stack Developer.

With the current end deliverable in mind, I am excited to get started. If this was something the college was also interested in, it could serve as a foray into feasibility.

## *Background (and References)*

My aim since the summer was to create a system composed of a web application, a database, and a system monitoring platform. On my internship at Mastercard, I was working with the Site Reliability Engineering team, where I created a similar distributed system. With the issues regarding registration and then Brightspace, the idea of creating a portal / hub, with Brightspace functionality, personalised for TUD, really interested me.

A large number of VLEs exist, with a number of different choices used in Ireland, for example, pointing to the fact that each university has distinct needs, like Brightspace, which we use in TUD [1] (also used in UCD [2]), with competitors such as Moodle (used by DCU [3] and Maynooth [4]), Loop (also used in DCU [5]) & Blackboard (TCD [6]) being used by other universities in the Republic of Ireland. An example outside the country would be Canvas (Oxford University [7]).

Portals / Hubs for universities are common, with their systems combined into one, removing the necessity for those at the relevant college to move between multiple resources, like St. Louis University in the US [8]. While the TUD website has the links to the various services available, it could be served with an option to log in at the top, and from there the students and lecturers can access a portal relating to their studies.

As an interesting point, UCD are also seemingly doing extensive research into developing a support structure for VLE Standards [9].

With a long list of existing VLEs and Portals that have been designed to either work internally to suit their own university's unique needs, or as a widely available customizable platform that a business model can run on, there are plenty of potential competing solutions that I can look towards for inspiration, to see what is the required competitive level of features / accessibility / availability and for what areas I can research.

As a student at the college, I am familiar with the currently available services, and I interact with people on a daily basis that also use them, which will be very helpful in trying to design a system worthy of meeting the necessary requirements.

I do not aim to replace the website, or the available Office365 services, I hope to incorporate them into my design as well.

## *Proposed Approach*

For my proposed approach in bringing this project to fruition, I will lay out the following plan for completing my project.

### **Research**

The first stage of this project is research. The first initial research point is the feasibility of the idea, can I do it with the time I have within reason, the technologies at hand, and the skills I have. Once I decided on this project idea, the next stage is researching deeper, in the following way.

#### ***The Framework***

This is the foundation for my project, inquiring into what are the core technologies I require to design this project, discovering if or how they will work with one another, starting from the bottom up: Virtual Machine Deployment until Fully Connected & Functional System. I will be asking some of the following questions:

- What virtual machine software will I use?
- What operating systems will my virtual machines have?
- What technologies for building web applications, databases and monitoring systems support these operating systems?
- Will those technologies work in tandem with one another?

This will give me the know-how for how exactly I will begin designing and implementing my project.

#### ***The Details***

This will be the second stage of my research, where I will research the important real-world problems that may be associated with a program such as this. Areas such as security, data integrity for GDPR related laws, accessibility and more, will be crucial for providing the best implementation that I can come up with, to hopefully lead to a feasible real-world design. I will be asking some of the following questions:

- How should I secure my network?
- How should a database be designed to stay in line with GDPR laws?
- What should I offer as features for accessibility, such as changeable colour schemes for colour blindness?

#### ***The Implementation***

This is where I begin the final stage of research, with the implementation. Now that I know what I have to base my physical implementation on, I can research deeper on how I should implement the project using the previously researched areas, starting with the web application. I will be looking for the best frameworks / modules for visual design, while looking to ensure features remain feasible when working with my chosen database. For the database, I will need to know how I will design the layout, in order to work with this web application and its features, including what are the necessary tables and queries that I will need to prepare for storing data and providing functionality, and what I need to keep in mind for a secure implementation in relation to requesting data. For the monitoring system, I will look into what are the best metrics to record and measure the availability of the system. Details such as these will be essential for my design and implementation.

## **Requirements Gathering**

Here is where I begin laying out what is needed to meet the standard and level necessary for this system to function. If I am going to be replacing Brightspace and the registration system, I need to know what should be kept, removed or added. By forming a questionnaire and a user group with a complex variety of needs, I can form the foundation for the design of my system.

For my user group, I have a number of people in mind that I have asked already to help, with different backgrounds, and a variety of needs, including former students, students from other colleges, people with dyslexia and autism, and potential mature students. The more direct feedback I can get, the more it will help bolster the research I will do, to account for as many backgrounds as I can.

## **Analysis & Design**

Once I know what I need, would like to implement, and what to avoid, I can begin planning the design to implemented, while analysing it for where I can improve the project.

I will first be designing the pages of the web application, the framework of the application. After, I will then begin laying out a database design to accommodate the features of the application, comparing the design implementation needed between the web pages and the database. Once I have designed my web page setup, and how it will interact with the database design, I will then design the monitoring system, analysing the most important parts of the system that will need to be monitored, and how I will implement the monitoring.

## **Implementation**

Once the analysis and design are finished, I will begin turning it into code and material work. Starting with setting up the virtual machines and the local network, I will then implement the web application in tandem with the database to check functionality as I go, before I finally setup the monitoring system, ensuring that the connection between one another is stable and accessible.

## **Testing / Evaluation**

Once my implementation is complete, I will begin my testing and evaluation of my project, starting with my own self-testing, knowing what weaknesses could exist, what was tricky to implement and very possibly done incorrectly. I will then ask members of my user group to use the program themselves, to discover any more bugs that may have been missed. Once I have gotten all this feedback, I will evaluate, then implement or change what is necessary.

## Deliverables

For the interim, I will be aiming to deliver a functional prototype with some feature implementation, consisting of a web application supported by a database, deployed on a local network. This will be delivered with custom user data inside my database, fabricated to mimic that of real student data. I plan on using Minikube with VirtualBox to deploy the local system on a single device, mimicking a design to be deployed later on the cloud as a functioning Kubernetes cluster. I will also be aiming to deliver an interim report, relating to the initial design and implementation, for this prototype.

For my final deliverable, I will deliver a fully functional web application, supported by both a database and monitoring platform, forming a distributed system, deployed and accessible through the web. I will also be aiming to deliver a comprehensive and structured report, filled with the necessary details for research, design and implementation, accounting for, scalability, reliability, accessibility, security, and accounting for real-world issues, such as the strict guidelines outlined in the General Data Protection Regulation (GDPR).

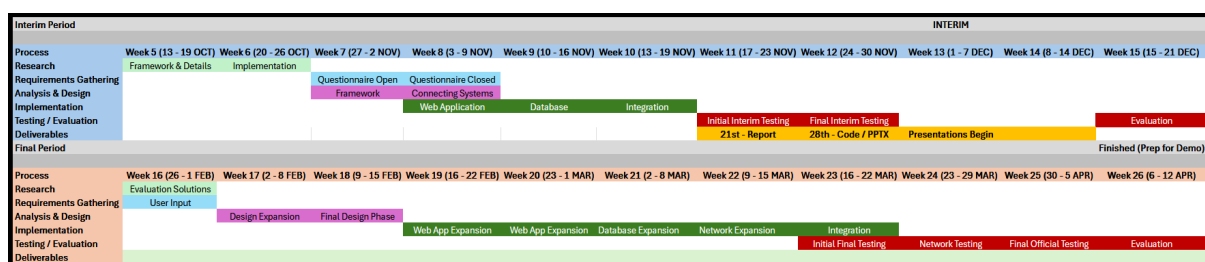
## Project Schedule

My schedule is based on the given timeline we have received, and I have broken it down into rough weekly parts where I will know what I need to have done in time to finish the project, potentially adjusting where necessary.

There are two main periods – one aiming for the Interim Report / Demo, while the latter will expand on the functionality and depth of the system, for the final version of the project.

The areas are based on my previously outlined approach, with an aim to have finished my development cycle by early to mid-April, adjusting where necessary once the final deadline has been confirmed. The aim for my final report is to write it once the document has been provided, filling it in bit-by-bit while working on the project.

### GANTT Chart For Project Schedule



See Blown Up Rotated Version in Appendix D



## *Technical Requirements*

The technical requirements of this project consist mainly of software and infrastructure, with a need for front-end and back-end technologies for my web application, supporting by caching and a database technology. With the goal of also implementing a platform monitoring service, to measure reliability and availability, technology will be needed for that as well.

My plan for my software technology stack includes HTMX for my front-end, Django for my back-end, Redis for caching and PostgreSQL for my database. With Minikube as my initial infrastructure technology for deployment on my local machine, I have the aim of deploying a Kubernetes cluster on the cloud down the line, depending on my options financially. If not, the system will be designed to be deployable on the cloud. In regards to hardware, this system will be initially deployed on my laptop.

## *Conclusion*

In conclusion, this project has the aim of tackling the issues faced this summer by the college, by replacing Brightspace and the attached systems, in order to create a streamlined, sustainable and reliable alternative, one that the people who study and work here, could have a voice in.

The project is ambitious, and should there be places in this project where I cannot fill the necessary gaps as a single developer, I will provide thorough research and reporting on what could be done to improve that area, why it should be considered, and how it could be done.

It is a deep dive into a full-stack distributed system, composed of a web application, database and monitoring – aiming for the professional standard that will be found in the workforce, and worthy of the time and effort I have spent in studying this course.

## References

- [1] T. U. Dublin, 'Connect | VLE - Brightspace | TU Dublin'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.tudublin.ie/connect/vle/>
- [2] 'UCD Connect'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.ucd.ie/connect/>
- [3] 'Moodle | DCU'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.dcu.ie/students/moodle>
- [4] 'Moodle | Maynooth University'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.maynoothuniversity.ie/centre-teaching-and-learning/technology-enhanced-learning/moodle>
- [5] 'About Loop | DCU'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.dcu.ie/teu/loop>
- [6] 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/>
- [7] 'Canvas | University of Oxford'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.ox.ac.uk/students/academic/guidance/canvas>
- [8] 'Using mySLU'. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.slu.edu/its/using-myslu.php>
- [9] '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/>

## *Appendix A: First Project Review*

<https://library.tudublin.ie/search?/Xtu856&SORT=D/Xtu856&SORT=D&SUBKEY=tu856/1%2C191%2C191%2CB/frameset&FF=Xtu856&SORT=D&30%2C30%2C>

<https://library.tudublin.ie/articles/5683162.6281/1.PDF>

### **Title**

Suaimhneas: A responsive web application for anxiety management and wellbeing

### **Student**

Ben McCormack

### **Description**

Web application designed to help people manage anxiety using CBT techniques, recording their daily moods, with personalized feedback and a recommendation system.

### **What is complex in this project**

His research is complex, diving deep into both the technologies he should use, and the experience for the user, and given that it is similar in setup to my own idea, in relation to the technologies and system he built, what he has done here could be a great help to understanding the level I require. While it is not relevant to my 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, and not really something I was familiar with.

### **What technical architecture was used**

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)

I will be looking into similar technologies, and when I was reading about Django, it was something I was interested in, as I love Python.

### **Explain key strengths and weaknesses of this project, as you see it**

Overall, this project seems to be on the stronger side, in most regards. His research was complex, thorough, the basis for his design and implementation on this research was equally as impressive. The main weakness I would say is his 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).

## *Appendix B: Second Project Review*

<https://library.tudublin.ie/articles/2945901.1871/1.PDF>

<https://library.tudublin.ie/search?/XDT228&SORT=DZ/XDT228&SORT=DZ&extended=0&SUBKEY=DT228/101%2C151%2C151%2CB/frameset&FF=XDT228&SORT=DZ&119%2C119%2C>

### **Title**

An eLearning system for Anonymous Feedback Sessions

### **Student**

Daniel Hogan

### **Description (brief)**

An older project, this was designed as an eLearning system that would provide the ability to give Anonymous Feedback,

### **What is complex in this project**

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.

### **What technical architecture was used**

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.

### **Explain key strengths and weaknesses of this project, as you see it**

There are more weaknesses in this project than the last. Despite the idea being more similar to my own than the first project, that project has a far more expansive implementation, one that I would be aiming to meet. 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 I will have to try and avoid.

Given that it is older, I would have to give leeway 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 I reviewed.

## *Appendix C: Prompts Used with ChatGPT*

N/A - Not Used In The Process

Appendix D:

Deliverables GANTT Chart

