



## LabLine - Final Delivery

BSC in Computing for Business  
4th Year - CSC1118  
2024/2025

| Name                           | Email  | Student Number |
|--------------------------------|--|----------------|
| Harsh Saini (Student)          | <a href="mailto:harsh.saini3@mail.dcu.ie">harsh.saini3@mail.dcu.ie</a>   | 21350903       |
| Eoin Francis (Student)         | <a href="mailto:eoin.francis3@mail.dcu.ie">eoin.francis3@mail.dcu.ie</a> | 21353371       |
| John McKenna (Project Advisor) | <a href="mailto:John.McKenna@dcu.ie">John.McKenna@dcu.ie</a>             |                |

## Table Of Contents

### Glossary

1. Overview
  - 1.1 Executive Summary
  - 1.2 Management Report
  - 1.3 Motivation
2. Business Case
  - 2.1 Description of Our Service
  - 2.2 Value Analysis of LabLine
  - 2.3 The Proposed Market
  - 2.4 Key Value Propositions
  - 2.5 User Segments
  - 2.6 Key Resources, Costs, Revenue Streams and Activities
  - 2.8 Chosen Pricing Model
  - 2.9 Competitor Profiles and Benchmarking
  - 2.10 Competitive Advantage
  - 2.11 Three Year Financial Plan
  - 2.12 Financial Requirements
  - 2.13 Primary Research
  - 2.14 Secondary Research
  - 2.15 BPMN
3. Technical Delivery
  - 3.1 Functional Specification
  - 3.2 Software Architecture
  - 3.3 AWS Diagram
  - 3.3 Logical Database Design
  - 3.4 Sitemap Diagrams
  - 3.5 Logical Database Design
  - 3.6 Use Cases
  - 3.7 Data Flow Diagram
  - 3.8 Performance Requirements
  - 3.9 Working Prototype
  - 3.10 Technical Challenges
  - 3.11 Source Code Highlights
  - 3.12 Interface Rationale

## Appendix

- A.1 Team CVs .
- A.2 Business Model Canvas
- A.3 Value Proposition Canvas
- A.4 Financial Overview
- A.5 LabLine Instructions For Students
- A.6 LabLine Instructions For Lab Tutors
- A.7 Financial Requirements

## Glossary

**AWS** - Amazon Web Services, a cloud platform where we hosted LabLine for user testing.

**QMS** - Queue Management System.

**KPI** – Key Performance Indicators

**ASGI** – Asynchronous Server Gateway Interface, A django interface standard that supports asynchronous features like websockets, allowing real-time communication

**CSRF** – Cross site Request Forgery, A security feature in web applications that ensures form submissions are not forged

**EC2** – Elastic Compute Cloud, a cloud service provided by AWS that allows hosting scalable virtual servers

**RDS** – Relational Database Service, AWS standard database separate from the internal system database

**Websocket** – A protocol that enable two way communication in real time between users without refreshing the webpage

**QR Code** – Quick Response Code

**LMS** – Learning Management System, A platform that provides all necessary tools for learning such as timetables, module content etc.

**UI/UX** - User Interface / User Experience, The information displayed on the user's screen and how it guides the user across our system.

**BPMN** – Business Process Model and Notation, to model the processes occurring within our system

**Round Robin** -

**SQLite** – Django database to store user, request and lab layout information

## 1. Overview

### 1.1 Executive Summary

LabLine is a web-based application designed to streamline the process by which students in university computer labs request assistance from lab tutors. LabLine has been developed using Django and Django Channels, the system replaces the outdated and inefficient method of hand-raising with a digital queue that allows for real-time updates and smarter request management.

This system addresses the key pain points observed in lab environments: the lack of a structured and fair mechanism for managing student queries. Long wait times, student demotivation, and tutor inefficiencies were all common themes identified through primary research. By offering students a way to submit detailed help requests, which include their location and an option to describe the problem they are facing, our platform enables tutors to respond in an organised, efficient manner. It also reduces the cognitive load on tutors who otherwise struggle to manage multiple requests during peak times.

From a management point of view, the system offers the request history features for Tutors, Lecturers and the faculty in general which helps make informed decisions when scheduling and allocating resources, helping faculty optimise lab support. Features such as real-time notifications, and fair request distribution (round-robin tutor assignment), contribute to improved overall satisfaction.

The system was designed with scalability and maintainability in mind. We have hosted LabLine on AWS for reliable performance. It also offers flexible pricing models: a standard monthly licensing fee and a premium monthly subscription that includes advanced analytics and dedicated support. Our target market includes higher education institutions with frequent lab-based instruction, particularly within computing and engineering disciplines.

By leveraging modern technologies and an agile development approach, this solution provides tangible benefits for students, tutors, and university administrators alike. It represents not only an improvement in day-to-day lab operations but also a step forward in how educational support is delivered in the digital age.

### 1.2 Management Report

### 1.2.1 Current Organisational Structure

- We employed an agile approach to the project by conducting regular meetings between ourselves to discuss our progress and to address any issues we had. We would often meet in person to collaborate on the project but would also chat online when it was not possible to meet in person.
- We created a GitLab repo that has tracked our progress during the project and each of our contributions.
- We maintained a Trello board of our tasks to help assign due dates for each task and prioritise tasks.
- We also conducted meetings with our project supervisor to receive their feedback on our progress and who could advise us on how to approach our project. They were previously our lecturer for the Developing Internet Applications module where we were introduced to the basics of Django and were able to advise us on the Django side of the project.
- They also allowed us to demo our project in one of their scheduled lab sessions, which proves their dedication to our project.

### 1.2.2 Proposed Organisational Structure

#### Product/Project Manager

Oversees the entire development process of a project. Their responsibilities not only include ensuring the project is meeting its KPIs but also monitoring the market and other competitors to consider potential ideas.

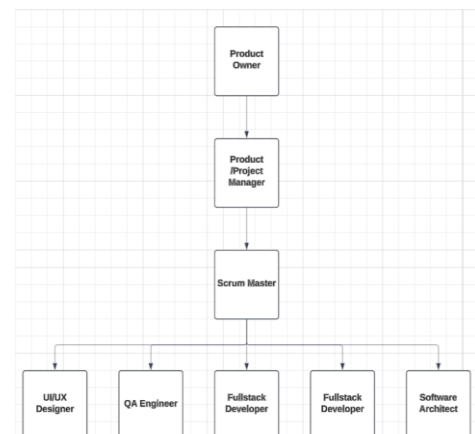


Figure1:ProposedOrganisationalStructure

#### Product Owner

Acts as a key link between the clients and the developers, allowing the developers to understand their end goals and to define their objectives.

## LabLine - Final Delivery

### Scrum Master

Apply scrum principles to projects to ensure the team is on track to complete the project. This is done through regular stand-ups and one-on-one meetings with team members. They use software such as Trello to maintain a backlog of future tasks.

### UI/UX Designer

UX designers research the behaviours of the target users to understand the UX that suits them. UI designers work with those in UX to create a UI that is intuitive and accessible to ensure users can navigate our application comfortably.

### Quality Assurance (QA) Engineer

Uphold the product quality of a project. They test the system through the lens of a user to identify if the system meets quality standards and if it is fully usable. For LabLine, they could register as one of the user types and interact with others.

They then compile bug reports to be delivered to developers which describes the actions they took to encounter a bug, ensuring the developer fully understands the issue.

### Software Architect

Organises the internal arrangement of the software to align with the project requirements. They help find the most optimal technical solutions to align with the project's requirements.

### Fullstack Developers

Develop and maintain both the front-end and back-end software.

[1]

## **1.2.3 Team CVs**

Refer to section A.1 in the appendix.

## **1.2.4 Team member logs**

|              |   |
|--------------|---|
| Eoin Francis | <a href="https://docs.google.com/document/d/1K-_XJIFCkVRKNDQCXpBPaHE3KPea3WnHD0pnO2TlvFA/edit?tab=t.0">https://docs.google.com/document/d/1K-_XJIFCkVRKNDQCXpBPaHE3KPea3WnHD0pnO2TlvFA/edit?tab=t.0</a> |
|--------------|---|

|             |   |
|-------------|---|
| Harsh Saini | <a href="https://docs.google.com/document/d/1drUdVdYVri3ZBe6jGe2xNdrGWsgzPmg8CMmNbYmjMkl/edit?usp=sharing">https://docs.google.com/document/d/1drUdVdYVri3ZBe6jGe2xNdrGWsgzPmg8CMmNbYmjMkl/edit?usp=sharing</a> |
|-------------|---|

### 1.3 Motivation

The motivation behind LabLine comes from the challenges we and our peers have encountered repeatedly in traditional computer lab environments. As students who have participated in countless lab sessions over the course of our academic journey, we have had first hand experience of the frustration experienced during these labs when trying to ask for help. The method of raising a hand to request for help is outdated and inconvenient.

We created Labine to solve these very real problems for the students currently still attending university and the ones who will come after us. We wanted to provide students with fairness when asking for help and replace the uncertainty and hesitation with clarity and confidence. Something which is often overlooked is how the traditional methods fail the students who suffer from social anxiety or other barriers to participation. In many cases students hesitate to raise their hand or speak up in front of a group out of fear of judgment or embarrassment. Some students may even choose not to attend labs at all, knowing the discomfort that comes with having to publicly request help. We believe this is a serious issue and one that should never be overlooked. LabLine provides a discreet and digital method of requesting help and by using LabLine students can ask for help without drawing attention to themselves. This offers a more inclusive and supportive environment, in which every student, regardless of their ability can ask for help. With LabLine we hope to pave the way for future learning spaces to be more efficient and also more compassionate and student focused.

## 2. Business Case

### 2.1 Description of our Service

LabLine is a digital platform that facilitates structured, real-time communication between students and tutors during university lab sessions. The core functionality centers around a web-based queue where students can submit help requests and tutors can accept and manage them efficiently.

Each student can log in, select their lab PC number, describe their issue, and submit a help request. The system queues these requests and displays them on tutor dashboards, sorted by

submission time or, optionally, by a fair distribution mechanism such as round-robin assignment. Tutors can accept, resolve, or update requests, with live updates sent to both parties through WebSocket technology powered by Django Channels.

For students, this eliminates the need to physically raise their hand and wait indefinitely, allowing them to continue working while in the queue. For tutors, it provides an organised list of tasks, helping them stay on track even during busy lab sessions. The interface is designed to be intuitive and responsive, requiring minimal training for new users.

The system supports core features such as role-based access (student/tutor/lecturer), help request creation and management, real-time notifications, request status tracking, and cancellation functionality. It is deployed using AWS infrastructure for reliability and scalability, incorporating EC2 instances for the application backend, RDS for the database, and Route 53 for DNS management.

Additional features include an analytics module for faculty and lab coordinators to assess lab usage patterns, optimise staffing, and make data-driven decisions. A planned FAQ and knowledge base will further reduce repetitive tutor interactions.

The product is designed to integrate easily into existing university IT environments and is suitable for deployment in a wide range of academic institutions, particularly those offering computing and engineering courses which would have high levels of lab usage.

## 2.2 Value Analysis of Product

LabLine is designed to serve multiple stakeholders within different sectors, primarily the educational institutions. The primary users in an educational setting are students and tutors, while the secondary users include lecturers, and university administration. Each of these groups benefits from the system in the following ways explained below.

### **Primary Users**

Students are the core users of LabLine and in traditional lab environments they often face long wait times, confusion over the availability of the tutors and inefficiencies in getting the help they need. LabLine addresses these issues in the following ways:

Eliminates frustration from uncertainty - by displaying their exact position in the queue and an estimated waiting time, paired with live updates and notifications for every request students no longer have to wonder or be stressed about when they will receive help.

- Ensure Fairness - LabLine operates on a first come first served basis, which prevents any queue jumping and favoritism. In some cases, students may be waiting for the tutor to come to them but the tutor may be stopped by another student which leads to tutors not



being able to track who needs help as they are constantly approached by students. LabLine ensures that every student gets an equal opportunity to receive help and the queue stays intact.

- Another issue faced by students due to long wait times is demotivation. Since the time students spend waiting to be assisted during labs is so long they are demotivated to request for help and in some cases not to attend labs as a whole since they can complete the tasks at home or outside the lab at their own pace.
- Reduce classroom disruptions - Instead of students raising their hands or calling out for assistance, LabLine provides a silent, digital way to request help, keeping the learning environment more orderly.
- Encourages students to seek help - In a lot of cases students are afraid to ask for help, either because they feel they are asking “Stupid questions”, or they are too afraid to raise their hand or be seen due to their social anxiety, LabLine caters for these scenarios by being fully digital, meaning students don't feel afraid since only the tutors and the students themselves can see what they need help with and requesting help is only a mouse click away.
- Encourages independent learning - since students can see the queue, they can focus on their work and continue to attempt troubleshooting the issue themselves or start on another part while they wait since they don't have to keep their hand raised constantly waiting to be noticed. This also encourages students to develop their problem solving skills and learn independently.

Tutors play a crucial role in assisting students, but managing multiple requests manually can be overwhelming. LabLine streamlines their workflow by organising and keeping track of every request for them, in an easy to manage system. Tutors benefit with LabLine for the following reasons:

- Structured workload management - Tutors receive the help requests in a clear, ordered format, allowing them to prioritise and assist the student efficiently. This prevents the chaos of multiple students seeking help at once
- Even distribution of requests - In traditional settings, some tutors may be overwhelmed with requests while others remain underutilised. LabLine automatically assigns requests, balancing the workload fairly among all available tutors.
- Real time notifications - Tutors are instantly alerted when a new help request is submitted. This ensures they stay informed and can respond promptly, improving overall response times.
- Ability to manage requests effectively - tutors can accept, and complete requests as needed, ensuring that every student receives appropriate help, even if the original tutor is unable to assist.

## Secondary Users

Lecturers and Faculty derive strategic value from LabLine in the following ways:

- Data driven Lab management - LabLine tracks help request volumes across labs, allowing staff to identify peak periods and adjust tutor staffing accordingly.
- Advanced Performance metrics - request analytics, highlight student pain points and recurring issues, giving lecturers valuable insights into course comprehension and support need.
- Cost efficiency - by enabling smarter scheduling and identifying under/over utilised resources, LabLine can help reduce unnecessary tutor hours while ensuring support is available when its needed.

LabLine also goes beyond individual users, and help academic institutions achieve broader goals in the following ways mentioned below:

- Enhanced student satisfaction - quick and fair access to help promotes a positive, inclusive academic environment and improves retention.
- Modernisation and innovation - by adopting a scalable, cloud hosted system educational institutes demonstrate commitment to digital transformation and quality of teaching.
- Compliance and accessibility - digital systems are easier to monitor and adapt for students with accessibility needs, aligning with diversity and inclusion goals.

## 2.3 Proposed Market

Initial research suggests that there are approximately 50,000 universities around the world. This provides us with a very large Total Addressable Market (TAM). These universities provide education to an estimated 254 million students globally. However, universities are more concentrated in certain countries/regions than others. **[2]**

UNESCO has also found that the number of people in third level education has more than doubled between 2000 and 2020. This shows that third level institutions need to invest in technologies that can cater to the learning and need for support from more students.

Our Serviceable Obtainable Market will include universities in North America and Europe since they have the highest participation ratio in tertiary education. **[3]**

Our target market, particularly in our first three years as a company, will be universities located in Ireland.

We have found a total of 28 recognised universities in Ireland. We then refined this number to 11 relatively large universities with between 14,000 and 34,000 students based on the fact that DCU and several universities had roughly 19,000 students or slightly below that while a few had over 30,000. [4]

Another issue affecting universities is the increasing student to staff ratios. In Ireland for example, student-staff ratios are 23:1 compared to 13:1 in the 1980s. This puts further pressure on academic staff to address the needs of students. [5]

According to Zendesk, nearly 60% of customers cited long hold/wait times when interacting with an agent as the most frustrating aspect of bad customer service. LabLine can help with this since it can allow each agent to select requests matching their expertise. Status updates can also notify the customer of their position in their respective agent's queue, helping to ease their frustration with waiting. [6]

## 2.4 Key Value Propositions

LabLine is not just a convenience for students and tutors, it's also a strategic investment for educational institutions seeking to improve the quality, efficiency, and accountability of lab-based learning environments. Below are the key value propositions for the decision-makers and paying customers responsible for adopting and funding educational technology.

- Improved student satisfaction and engagement - By reducing the wait times and improving the support experience, LabLine improves student satisfaction and engagement, both of which are the key drivers of academic success and retention. Happier, better supported students are more likely to attend labs and engage with the module, perform better, and remain enrolled. This contributes to the positive institutional metrics.
- Data driven decision making - LabLine also provides actionable analytics on lab usage, request volumes, peak times and tutor performance. This data empowers faculty heads and department managers to make informed decisions about tutor staffing, resource allocation, and timetable design, leading to more efficient use of institutional resources.
- Optimised tutor allocation and cost savings - LabLine helps balance tutor workloads through automatic request distribution (e.g., round-robin), minimizing downtime and overload. With visibility into high- and low-demand periods, institutions can adjust staffing levels, potentially reducing the need for excess tutor hours — a direct cost-saving measure.
- Support for learning equity and inclusion - By removing social and psychological barriers (e.g., fear of asking questions), LabLine encourages all students — including those with social anxiety or learning differences — to seek help. This aligns with institutional diversity and inclusion goals and supports more equitable learning environments

## 2.5 User Segments

LabLine is designed to support a diverse but interconnected set of user segments within educational institutions. We chose these segments based on direct research, and our personal experiences in dealing with inefficiencies in current lab environments,

### 1. University Students

Students are the main beneficiaries of LabLine, as they are the ones seeking timely academic support during their computer lab sessions. Traditional methods of seeking help, such as hand-raising methods often leave students waiting too long for help, causing frustration, demotivation, and reduced engagement. LabLine addresses these issues with a structured, fair, and transparent help request system that allows them to focus on learning rather than competing for attention.

Use Cases:

- Submitting and tracking help requests during lab sessions
- Receiving real-time notifications when a tutor is available
- Viewing queue position and expected wait time

### 2. Lab Tutors

Tutors play a major role in supporting student learning in labs. However, they currently face problems in managing help requests. Without a formal system, it's hard to prioritise students, balance workloads, or track which requests have been resolved. LabLine gives tutors a streamlined dashboard to efficiently manage requests, helping them assist more students with less confusion and stress.

Use Cases:

- Viewing and managing incoming help requests
- Accepting and noting requests as completed
- Receiving automatic request assignments (e.g., via round-robin)

### 3. Lecturers

Lecturers also benefit from LabLine through higher student engagement and reduced lab disruptions. Lecturers can also access analytics that help them identify common problem areas faced by students to make more accurate decisions on the level of content being taught to them, assess tutor response rates, and optimise lab support. This data provides valuable insights for course planning and performance reviews.

### Use Cases:

- Reviewing aggregated request data and common topics of confusion
- Identifying lab sessions with high support demand
- Enhancing teaching strategy based on student behavior insights

## 4. University Administrators and Faculty

These stakeholders are responsible for budgeting, resourcing, and implementing educational technology within departments. LabLine offers them operational efficiencies, cost-saving opportunities, and data-driven insights that help optimise tutor allocation and support planning. By improving student support infrastructure, LabLine contributes to institutional goals such as retention, inclusivity, and teaching quality.

### Use Cases:

- Monitoring tutor performance and student engagement trends
- Using analytics to adjust tutor scheduling and lab coverage
- Aligning LabLine with digital transformation or EdTech strategies

## 2.6 Key Resources, Costs, Revenue Streams And Activities

### 2.6.1 The key resources

This section provides the breakdown of the key resources, operational activities, expected costs, and revenue streams of LabLine.

#### Human resources:

- Customer success and support team: This team is responsible for answering questions, resolving issues customers(Faculty managers in the Universities) face when using LabLine, processing returns or refunds, providing assistance via email, phone or social media, along with assisting new clients with set up and training.
- Development team: This team is responsible for the full stack development, WebSocket logic, performance optimisation, bug resolution and feature enhancements.
- Marketing and growth team: This team is highly skilled in digital marketing, they are responsible for the design and execution of marketing campaigns to increase the awareness of LabLine through various social media channels. They work consistently to push the presence of our product targeting both students(to show there is a solution to the problems they face) and universities.

### Technical Resources:

- Cloud infrastructure(AWS): LabLine relies on AWS services for hosting and scalability, this includes the following services/components:
  - EC2 for hosting the Django web application
  - RDS (PostgreSQL) for secure and reliable database storage
  - Route 53 for domain name management and routing
  - CloudWatch for system monitoring and performance metrics
  - Firewall to protect against malicious attacks such as SQLi (an attack involving executing malicious SQL statements in a database).
  - Load balancing to provide scalability at times of high usage of LabLine by evenly distributing internet traffic across multiple EC2 instances.
- Django: LabLine has been built using Django and Django channels with which the system handles real time communication(Websockets) for live updates between the users for the notifications, status updates, request creation and request deletion
- Version control and Continuous integration/Continuous deployment: Gitlab will be used for collaborative development, code reviews and continuous integration/deployment.

### **2.6.2 Overview of the costs**

#### Marketing

The marketing expenses will focus on attracting new customers and retaining existing ones. These costs are related to digital advertisements, social media campaigns, and other promotional activities. For LabLine strategic marketing initiatives are mandatory for growing the brand, and establishing a strong lasting impression. How LabLine is marketed is very important as it will help attract the correct customers to our product and help bring in more sales.

#### Digital Infrastructure

Digital infrastructure expenses are related to the development, and maintenance of the technologies supporting LabLine. This includes the costs for hosting the application on AWS, Database management, cloud services and ongoing software development. Investing in a digital infrastructure that is scalable is highly important to ensure that LabLine is reliable, secure and functions perfectly.

#### Employees

The Employee costs cover salaries, and other expenses paid by employees(e.g. Travel expenses), it is important we hire and maintain dedicated employees as they will be responsible for the software development, customer support, marketing, and other administrative tasks. By investing in a competent workforce we can see better results and customer satisfaction.

### **2.6.3 Revenue Streams**

#### Standard monthly licensing (Core Plan)

The monthly Licensing plan serves as the foundational revenue stream for LabLine, it is designed to offer a cost effective way to integrate the system into the lab environments without requiring complex financial commitments. We have priced this at €1,500 per lab per month, and this plan includes access to LabLine's core features, such as the digital help request queues, dashboards for each user type, real time notifications, and basic usage analytics. This pricing model is particularly attractive to smaller departments or universities piloting/trialling the system, as it provides a predictable cost structure with minimal administrative overhead. By offering monthly billing LabLine aligns with how most universities handle procurement and IT spending

#### Monthly subscription (Premium Plan)

The monthly subscription plan complements the licensing model by offering institutions access to a more robust, feature rich version of LabLine for €400 per lab per month. Aimed at high demand labs, this plan unlocks the advanced analytics, including usage trends, tutor response times, and historical request summaries, it also includes priority access to new features and platform updates. In addition to this, institutions on this plan can request custom branding, and lecturer level dashboards. The flexibility of a monthly subscription allows universities to scale up temporarily during exam periods or lab intensive semesters, and then scale back if needed. The monthly subscription plan offers customers the option to only pay for the capacity and functionality they truly need at a given time.

### **2.6.4 The Activities required**

#### Development and Feature expansion

At the heart of LabLines continued value is ongoing commitment to product development. This includes maintaining the core system stability while introducing new features based on user feedback, competitive analysis and evolving educational needs. Activities involve regular sprint cycles, software updates, UI improvements, and enhancements such as queue automation logic, or deeper analytics capabilities based on the needs of the users. Continuously improving the product ensures it remains relevant and responsive to changing academic environments and technological expectations.

#### System maintenance and Infrastructure Maintenance

To support a smooth, scalable, and reliable experience for all users, regular system maintenance is critical. This includes monitoring AWS server usage, optimising database performance, handling backups, patching security vulnerabilities, and managing real time websocket communication established with Django channels for all users. Infrastructure management also involves scaling the system for increased load during peak academic periods, ensuring high

uptime and responsiveness. These activities are essential for building trust with institutions and guaranteeing operational integrity.

### Marketing

Our marketing plan will initially focus on acquiring and attracting new customers to our business, we will aim to ensure that every university and student is aware of our product and its importance. We will mainly focus on social media to make our business known to the correct customers, with social networking websites like LinkedIn and Instagram since they are the most effective in reaching the University faculty members and students. Another way we intend to promote our business is by conducting demonstrations of LabLine in educational institutions, to show a visual representation of the different features, the effectiveness and improved productivity in lab sessions.

## 2.7 Supporting Industry Trends

The Edtech market which would broadly include our value proposition and since it focuses more on educational institutions including universities, is expected to have a compound annual growth rate (CAGR) of 15.5% between 2023 and 2032. One of the key factors in the increase of the Edtech market is the adoption of technology to improve accessibility and engagement in education. This indicates that universities are more likely to allocate a larger portion of their budget towards innovative technology solutions. [7]

The number of enrollments in ICT related courses in Ireland increased from 15,590 in 2017/2018 to 21,195 in 2023/2024 according to the Higher Education Authority (HEA). This provides us with a growing user base where growth was only stagnant during the COVID 19 pandemic but resumed when it ended. This also means that computer labs will be used more frequently by students if these facilities cannot expand fast enough. This would require a way for the process of students asking for help during labs to be streamlined to handle the growing number of students studying in this area. [8]



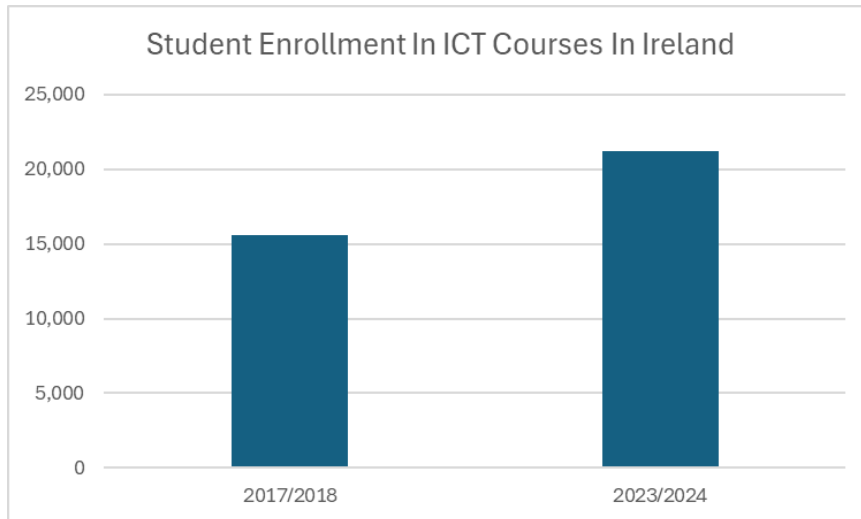


Figure 2. Growth in the number of students enrolled in ICT courses in Ireland between 2017/2018 and 2023/2024.

Student enrollment in Computer Science courses is also the fastest among any university subject in the UK at 7%. This represents an urgent need for a system such as LabLine to help handle the increased demand from students. **[9]**

There has also been increasing adoption of Learning Management Systems (LMS) in universities including Moodle and DCU's Loop platform.

A key emerging trend in LMSs is the convergence of technologies into one solution where students can access the tools they need from one LMS. Feedback from our primary research included that LabLine could be fully integrated into Loop once it is a fully developed system, supporting this trend towards a unified system.

We plan to integrate LabLine with Moodle as a key partner. Moodle had a market share of 69% in the European LMS market in 2023, a region which we previously identified as having a high participation rate in tertiary education. **[10] [11]**

We have also analysed the queue management system market since our system will utilise elements found in these systems. This market is expected to grow with a CAGR of 7.5% from 2023 to 2031. The virtual queue category of this market which aligns with the virtual queueing aspect of our system had the largest share within the queue management system market in 2023 and is expected to have the highest CAGR.

The real-time monitoring segment within the Queue Management System market is expected to have the fastest growth rate with a CAGR of 9.66% from 2024 to 2032. This involves businesses seeking to improve both operational efficiency and customer satisfaction through insights into customer flow. LabLine aligns with this trend since it provides immediate updates to students

when a tutor has accepted their request, ensuring that they don't feel their request has been forgotten.

LabLine's ability to provide maps of computer labs with the locations of students sending requests helps for tutors to be more efficiently relocated to labs where there is more demand. **[12]**

### 2.8 Chosen Pricing Model

LabLine's core plan costs €1,500 per lab per month, to reflect that it includes features such as lab support and lab management all in one place.

This monthly fee includes the full deployment of LabLine, fully comprehensive onboarding and training for tutors and lecturers, system hosting and dashboards tailored for user types such as students, tutors and lecturers. It also includes the core functionality for the most essential tasks including queue management and tracking request status. This is unlike traditional ticketing based systems such as Zendesk which cater to a more broad audience.

An additional subscription of €400 per lab per month with the existing monthly fee will provide additional features including peak-time analysis, tutor performance analysis and historical request reporting. Priority support and early access to new features are also included to ensure our customers' concerns are addressed as soon as possible.

LabLine significantly reduces the administrative and overhead costs usually needed for busy computer labs by consolidating analytics, tracking student engagement and real-time tutor coordination.

LabLine also improves student accessibility, especially for those who may have social anxiety or learning differences who may shy away from asking for help in traditional computer lab settings. Through real time analytics and logs, institutions can gain key insights into tutor performance, student pain points and how busy labs are, ensuring that institutions can uphold their reputation. Compared to the combined costs of hiring extra lab assistants (especially undergrad and postgrad students who are already busy with their studies), manual analytics and the need for multiple software to conduct each of one of these tasks, LabLine delivers strong return on investment for institutions committed to improving the performance and learning experience of their students.

LabLine provides these needs in one place for just €18,000 per month. This means that once LabLine has been configured, features such as analytics, student engagement and a more organised workflow for lab tutors become available without the need to configure any other software and manage payments/subscriptions to multiple services.

There is also the need to pay the salaries of extra lab tutors when necessary during busy labs. The estimated annual salary of one lab tutor is approximately €30,000 per year which can be saved by ensuring that tutors already present are not overwhelmed by providing them with a structured list of requests through LabLine. [14]

### 2.9 Competitor Profiles and Benchmarking

While no notable competitors exist that directly address the same target market as LabLine, our main competitors include those in the queue management system market. On a broad scale, Zendesk provides many similar features to LabLine. It allows users to send requests in the form of tickets which are received by an agent. Users are then notified that their request was received by the support team, similar to how students are notified that their request is received and accepted by a tutor.

Zendesk offers its services in tiered subscriptions, with additional features and support at higher tiers. [15] [17]

Qless also provides a queue management system. It focuses more on queue management through features such as updating users when their waiting time is over rather than resolving requests. [16]

### 2.10 Our Competitive Advantage

LabLine combines features from both Zendesk and Qless. Similar to Zendesk, LabLine provides lab tutors with a dashboard to manage requests allocated to them but can also be adapted for in person environments.

LabLine also provides status updates to users through its UI, contrasting with ticketing systems such as Zendesk where notifications are provided via email, which the user may not necessarily be checking. This is especially true in a computer lab setting, where students are more focused on completing tasks than checking their emails regularly.

Zendesk and Qless are also targeted towards the customer service industry, meaning they may not be made to be integrated with LMS systems similar to LabLine.

There are no notable competitors who offer the exact services LabLine provides to universities. For example, some universities may use Qless to handle scheduling appointments while they may use SurveyMonkey to gather feedback from students.

### 2.11 Three Year Financial Projection

Refer to section A.7 in appendix for more details

Year 1

LabLine - Final Delivery

| INVESTMENT INCOME                    | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
|--------------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Angel Investment                     | 50,000.00  | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | 50,000.00  |
| AIB Loan                             | -          | -          | -          | -          | -          | 40,000.00  | -          | -          | -          | -          | -          | -          | 40,000.00  |
| Founder Investment - Harsh           | 25,000.00  | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | 25,000.00  |
| Founder Investment - Eoin            | 25,000.00  | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | 25,000.00  |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| TOTALS                               | 100,000.00 | -          | -          | -          | -          | 40,000.00  | -          | -          | -          | -          | -          | -          | 140,000.00 |
| OPERATING REVENUE                    | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| Monthly Fee                          | -          | 3,000.00   | 3,000.00   | 7,500.00   | 7,500.00   | 7,500.00   | 10,500.00  | 10,500.00  | 10,500.00  | 15,000.00  | 15,000.00  | 15,000.00  | 105,000.00 |
| Subscription Plans                   | -          | 400.00     | 400.00     | 1,200.00   | 1,200.00   | 1,200.00   | 1,600.00   | 1,600.00   | 1,600.00   | 2,400.00   | 2,400.00   | 2,400.00   | 16,400.00  |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| TOTALS                               | -          | 3,400.00   | 3,400.00   | 8,700.00   | 8,700.00   | 8,700.00   | 12,100.00  | 12,100.00  | 12,100.00  | 17,400.00  | 17,400.00  | 17,400.00  | 121,400.00 |
| MONTHLY EXPENSES                     | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| Salaries (dev)                       | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 84,000.00  |
| Salaries (Customer Support)          | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,000.00   | 48,000.00  |
| Salaries (Marketing and growth team) | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 14,400.00  |
| Marketing Expenses                   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 1,011.67   | 12,140.00  |
| Travel                               | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 3,600.00   |
| AWS Cloud Services                   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 14,400.00  |
| Technical Services                   | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 1,786.20   |
| Legal GDPR Consultant                | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 1,680.00   |
| Loan Repayments                      | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 14,945.64  |
| Misc                                 | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 3,000.00   |
| VAT (Sales Tax)                      | -          | 782.00     | 782.00     | 2,001.00   | 2,001.00   | 2,001.00   | 2,783.00   | 2,783.00   | 2,783.00   | 4,002.00   | 4,002.00   | 4,002.00   | 27,922.00  |
| BUPIN RATE                           | 16,496.39  | 17,277.99  | 17,277.99  | 18,496.39  | 18,496.39  | 18,496.39  | 19,278.99  | 19,278.99  | 19,278.99  | 20,497.99  | 20,497.99  | 20,497.99  | 225,873.84 |
| OVERVIEW                             | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| CUMULATIVE EXPENSES                  | 16,496.39  | 33,773.97  | 51,051.96  | 69,548.95  | 88,045.93  | 106,542.92 | 125,821.91 | 145,100.89 | 164,379.88 | 184,877.87 | 205,375.85 | 225,873.84 | 225,873.84 |
| CUMULATIVE INCOME                    | 100,000.00 | 103,400.00 | 106,800.00 | 115,500.00 | 124,200.00 | 172,900.00 | 185,000.00 | 197,100.00 | 209,200.00 | 226,600.00 | 244,000.00 | 261,400.00 | 261,400.00 |
| STATUS                               | 83,504.01  | 69,626.03  | 55,748.04  | 45,951.05  | 36,154.07  | 66,357.08  | 59,178.09  | 51,999.11  | 44,820.12  | 41,722.13  | 38,624.15  | 35,526.16  | 35,526.16  |

Figure 3.

Year 2

| INVESTMENT INCOME                     | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
|---------------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Angel Investment                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| AIB Loan                              | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| Founder Investment - Harsh            | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| Founder Investment - Eoin             | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
|                                       | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| TOTALS                                | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| OPERATING REVENUE                     | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| Monthly Fee                           | 18,000.00  | 18,000.00  | 18,000.00  | 19,500.00  | 19,500.00  | 19,500.00  | 25,500.00  | 25,500.00  | 25,500.00  | 31,500.00  | 31,500.00  | 31,500.00  | 283,500.00 |
| Subscription Plans                    | 2,800.00   | 2,800.00   | 2,800.00   | 3,200.00   | 3,200.00   | 3,200.00   | 4,000.00   | 4,000.00   | 4,000.00   | 4,800.00   | 4,800.00   | 4,800.00   | 44,400.00  |
|                                       | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
|                                       | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
|                                       | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          |
| TOTALS                                | 20,800.00  | 20,800.00  | 20,800.00  | 22,700.00  | 22,700.00  | 22,700.00  | 29,500.00  | 29,500.00  | 29,500.00  | 36,300.00  | 36,300.00  | 36,300.00  | 327,900.00 |
| MONTHLY EXPENSES                      | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| Salaries (dev)                        | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 7,000.00   | 10,500.00  | 10,500.00  | 10,500.00  | 10,500.00  | 98,000.00  |
| Salaries (Customer Support)           | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 72,000.00  |
| Salaries (part time Marketing and gro | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 1,200.00   | 14,400.00  |
| Marketing Expenses                    | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 2,732.50   | 32,790.00  |
| Travel                                | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 3,600.00   |
| AWS Cloud Services                    | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 1,500.00   | 18,000.00  |
| Technical Services                    | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 148.85     | 1,786.20   |
| Legal GDPR Consultant                 | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 1,680.00   |
| Loan Repayments                       | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 14,945.64  |
| Misc                                  | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 3,000.00   |
| VAT (Sales Tax)                       | 4,784.00   | 4,784.00   | 4,784.00   | 5,221.00   | 5,221.00   | 5,221.00   | 6,785.00   | 6,785.00   | 6,785.00   | 8,349.00   | 8,349.00   | 8,349.00   | 75,417.00  |
| BUPIN RATE                            | 25,300.62  | 25,300.62  | 25,300.62  | 25,737.62  | 25,737.62  | 25,737.62  | 27,301.62  | 27,301.62  | 30,801.62  | 32,365.62  | 32,365.62  | 32,365.62  | 335,618.84 |
| OVERVIEW                              | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV        | DEC        | YR TOTAL   |
| CUMULATIVE EXPENSES                   | 251,174.66 | 276,475.48 | 301,776.30 | 327,514.12 | 353,251.94 | 378,989.76 | 406,291.58 | 433,593.40 | 464,395.22 | 496,761.04 | 529,126.86 | 561,492.68 | 561,492.68 |
| CUMULATIVE INCOME                     | 282,200.00 | 303,000.00 | 323,800.00 | 346,500.00 | 369,200.00 | 391,900.00 | 421,400.00 | 450,900.00 | 480,400.00 | 516,700.00 | 553,000.00 | 589,300.00 | 589,300.00 |
| STATUS                                | 31,025.34  | 26,524.52  | 22,023.70  | 18,985.88  | 15,948.06  | 12,910.24  | 15,108.42  | 17,306.60  | 16,004.78  | 19,938.96  | 23,873.14  | 27,807.32  | 27,807.32  |

Figure 4.

Year 3

## LabLine - Final Delivery

| INVESTMENT INCOME                    | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV          | DEC          | YR TOTAL     |
|--------------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|--------------|--------------|--------------|
| Angel Investment                     | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| AIB Loan                             | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| Founder Investment - Harsh           | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| Founder Investment - Eoin            | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| TOTALS                               | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| OPERATING REVENUE                    | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV          | DEC          | YR TOTAL     |
| Monthly fee                          | 31,500.00  | 31,500.00  | 31,500.00  | 37,500.00  | 37,500.00  | 37,500.00  | 37,500.00  | 37,500.00  | 37,500.00  | 37,500.00  | 37,500.00    | 37,500.00    | 432,000.00   |
| Subscription Plans                   | 4,800.00   | 4,800.00   | 4,800.00   | 5,600.00   | 5,600.00   | 5,600.00   | 5,600.00   | 5,600.00   | 5,600.00   | 5,600.00   | 5,600.00     | 5,600.00     | 64,800.00    |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
|                                      | -          | -          | -          | -          | -          | -          | -          | -          | -          | -          | -            | -            | -            |
| TOTALS                               | 36,300.00  | 36,300.00  | 36,300.00  | 43,100.00  | 43,100.00  | 43,100.00  | 43,100.00  | 43,100.00  | 43,100.00  | 43,100.00  | 43,100.00    | 43,100.00    | 496,800.00   |
| MONTHLY EXPENSES                     | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV          | DEC          | YR TOTAL     |
| Salaries (dev)                       | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00  | 14,000.00    | 14,000.00    | 168,000.00   |
| Salaries (Customer Support)          | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00   | 6,000.00     | 6,000.00     | 72,000.00    |
| Salaries (Marketing and growth team) | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00   | 2,200.00     | 2,200.00     | 26,400.00    |
| Marketing Expenses                   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00   | 4,140.00     | 4,140.00     | 49,680.00    |
| Travel                               | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00     | 300.00       | 300.00       | 3,600.00     |
| AWS Cloud Services                   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00   | 2,000.00     | 2,000.00     | 24,000.00    |
| Technical Services                   | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39      | 82.39        | 82.39        | 1,340.56     |
| Legal GDPR Consultant                | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00     | 140.00       | 140.00       | 1,680.00     |
| Loan Repayments                      | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47   | 1,245.47     | 1,245.47     | 14,945.64    |
| Misc                                 | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00     | 250.00       | 250.00       | 3,000.00     |
| VAT (Sales Tax)                      | 8,349.00   | 8,349.00   | 8,349.00   | 9,913.00   | 9,913.00   | 9,913.00   | 9,913.00   | 9,913.00   | 9,913.00   | 9,913.00   | 9,913.00     | 9,913.00     | 114,264.00   |
| BURN RATE                            | 38,786.05  | 38,786.05  | 38,786.05  | 40,350.05  | 40,350.05  | 40,350.05  | 40,350.05  | 40,350.05  | 40,350.05  | 40,350.05  | 40,350.05    | 40,350.05    | 479,518.20   |
| OVERVIEW                             | JAN        | FEB        | MAR        | APR        | MAY        | JUN        | JUL        | AUG        | SEPT       | OCT        | NOV          | DEC          | YR TOTAL     |
| CUMULATIVE EXPENSES                  | 600,279.53 | 639,066.38 | 677,853.23 | 718,204.08 | 758,554.93 | 798,905.78 | 839,256.63 | 879,607.48 | 919,958.33 | 960,309.18 | 1,000,660.03 | 1,041,010.88 | 1,041,010.88 |
| CUMULATIVE INCOME                    | 625,600.00 | 661,900.00 | 698,200.00 | 741,300.00 | 784,400.00 | 827,500.00 | 870,600.00 | 913,700.00 | 956,800.00 | 999,900.00 | 1,043,000.00 | 1,086,100.00 | 1,086,100.00 |
| STATUS                               | 25,320.47  | 22,833.62  | 20,346.77  | 23,095.92  | 25,845.07  | 28,594.22  | 31,343.37  | 34,092.52  | 36,841.67  | 39,590.82  | 42,339.97    | 45,089.12    | 45,089.12    |

Figure 5.

## 2.12 Financial Requirements to bring Idea to Completion

Refer to section A.7 in appendix

To successfully bring LabLine from its current prototype stage to a fully operational, commercially available platform will require investments in development, infrastructure, deployment, marketing and support.

What features would you like to see added to LabLine in the future?

5 responses

To be able to add someone else in on your query

I would like to see if this could run during different hours of the day when people are doing lab work individually and need additional help.

maybe if you have the same question as someone else it could say and then the tutor could help the two students at the same time

just general UI improvements, see below.

-instead of cancelling I would like to be able to edit my request while staying in the line. An expected wait time. The TA that will be helping me.

Figure 6.

## Software development and Feature Completion

While the core functionality of LabLine is already operational, additional features based on the feedback we received after user testing(shown in the image on the right) must be developed to make the product more satisfactory for the end users.

## LabLine - Final Delivery

### Infrastructure and hosting set up

LabLine will continue to be hosted on AWS and will use the following Cloud services:

- EC2 instances for application hosting
- RDS PostgreSQL for database management
- S3 for file storage
- Route 53 for domain and DNS routing

These costs will cover setup, monitoring tools(AWS Cloudwatch), and a year's worth of hosting for the initial lab deployments. Infrastructure is scalable, with costs rising only modestly as we acquire more customers and add more labs.

### Marketing for LabLine

To acquire customers and make our product be known to the public we will have to conduct targeted launch campaign which will focus on:

- Outreach to educational institutions
- Attendance at universities to demo the product
- Market our product on social media platforms, like LinkedIn and Instagram

The capital used for marketing will cover running advertisements on social media, design tools to enhance/create our products logo and promotional materials.

### Salaries

To maintain smooth operation and maintenance of LabLine we will employ the following human resources:

- 2 Software developers
- 2 customer support and success staff
- 1 part time marketing/social media manager

## 2.13 Primary Research

To understand how users felt using our system, we carried out user testing to determine their opinion on LabLine and to gather their feedback to improve the system.

Before conducting this, we needed to decide on a method to gather their feedback after user testing was completed. We chose a feedback survey since this could help us benchmark the results of our system, allowing us to identify the key problems experienced by users and their suggested improvements.

We researched possible questions to include in our survey. We found that it was optimal for the questions to capture a broad overview of the user's experience, through areas such as their overall experience, the UI and their favourite features. **[18]**

We also researched surveys for similar systems to find which questions they included. **[19]**

## LabLine - Final Delivery

We chose to demo LabLine during a lab session for the Developing Internet Applications module since we could demonstrate to them an example of what could be developed for a final year Django application.

Results from our survey showed that most respondents had a positive experience with LabLine and would recommend it to others. They found the UI easy to navigate. The survey also found they did not need to keep their hand raised to receive help.

How likely are you to recommend LabLine to others?

5 responses

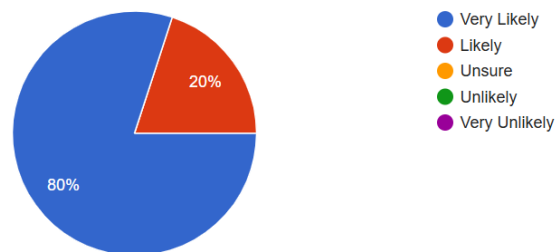


Figure 7.

How easy was LabLine's UI (User Interface) for you to understand and navigate?

5 responses

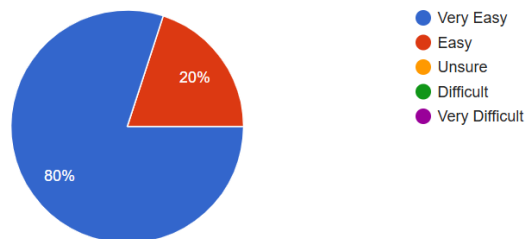


Figure 8.

Recommendations from students on how we could improve LabLine included the ability for students to group themselves together regarding a specific query. This correlates with a similar idea that was implemented in a case study in the University of Illinois-Urbana in the secondary research section.

This has helped to validate the need for LabLine within university computer labs since students expressed overall satisfaction with the system and recommended what changes should be made. This demonstrated that the students could see the potential for LabLine and they are invested enough to help us build on what we have already developed.

Tutor feedback from our user testing included that it was easier for them to see which students needed their help next. They found the UI very easy to navigate similarly to students.

A key piece of feedback was that they recommended there be a lab map to allow them to locate students easier. We were able to integrate this into our project where students who have pending requests can be identified by their closest PC which turns from green to red on the lab map. They also recommended an estimated wait time to identify if students have been waiting particularly long. We have also implemented this into LabLine since this can allow tutors to note if they have spent an excessive amount of time with one student and cannot allow a large backlog of pending requests to form.

### 2.14 Secondary Research

A case study was conducted in the University of Illinois-Urbana where students could use a queueing system online where they add themselves to the queue online. While mainly for students waiting to see an instructor in their office, students could indicate their request topic and location (e.g. Computer lab N, Seat 13), allowing the instructor to locate them.

The system was found to have saved time for both students and instructors. It also helped alleviate previous problems students faced.

Instructors could also identify which problems were most frequent among students, helping them focus on specific topics in more detail.

It also expanded to over a dozen courses in the university since it's implementation, proving it's popularity among students and staff.

Before, students had to pack into rooms and write their names on a whiteboard which interrupted their workflow. **[21]**

Another study in Linköping University involved the usage of Camedin, a web app that provides a live page from where students could send requests to teachers in a computer lab. Teachers were initially overwhelmed by the large number of requests from students to the extent that some labs were left unattended.

45 of the 150 students involved in the study were asked what effect they believed Camedin had on their lab experience. The results showed that the most referenced benefits were time saving, efficiency and simplicity, similar to the findings from our primary research. Students could also focus more time towards their work and not spend time searching for teachers in other locations. Much fewer students also complained about lateness/lack of assistance in the evaluation of the course at the end. **[22]**

The above case studies highlight how similar queueing systems have been tested and received positive feedback, proving that such systems can be beneficial for students and academic staff.

### 2.15 BPMN



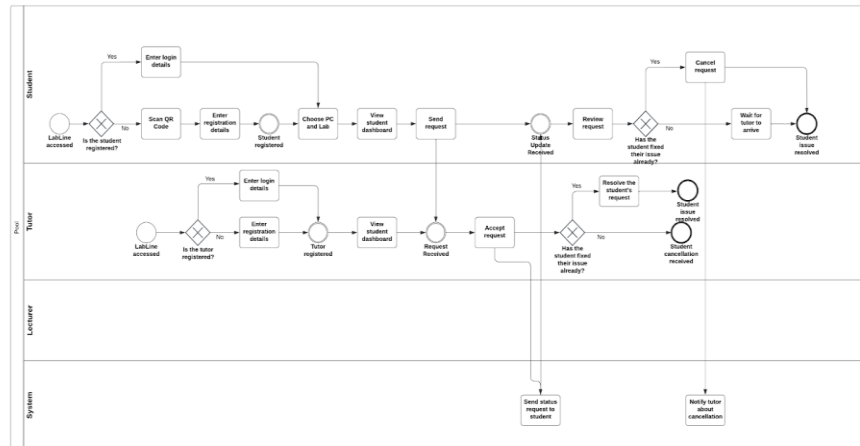


Figure 9. BPMN diagram for a student sending a request to a tutor

The BPMN above highlights the process of a student registering/logging into LabLine to send a request to a lab tutor.

1. Registration/Login: The user scans the QR code and registers if they have not already registered. If they are already registered, they will enter their login details.

Once registered/logged in, they choose their PC number and lab.

2. Student dashboard: After viewing the student dashboard, the student can send a request for help. This will be received by a tutor who will accept the request when ready. When the tutor accepts, the system sends a status update to the student notifying them their request is accepted.

The student can wait until a lab tutor resolves their issue, or if the student has resolved it already, they can cancel their request.

## 3. Technical Delivery

### 3.1 Functional Specification

#### **register and student\_register**

- Manages registration for users. Register.html manages registration for tutor and lecturer user types. Student\_register.html manages registration exclusively for students who scan a QR code. This ensures that they cannot access more administrative roles in the system and potentially disrupt the flow of lab sessions.

The QR code is included in section A4 in the appendix.

Procedure:

- Provide a registration form for new users.
- Scan the QR code in the appendix to access the student\_register view.
- Redirect the new user to their relevant dashboard or to choose\_lab\_and\_pc if they are a student.

#### **Login\_view and logout\_view**

- Handles logging in and logging out user sessions.
- Procedure:
  - Login\_view logs the user in and redirects them to the relevant dashboard based on their user type.
  - Logout\_view ends the user's session and redirects them to the login page.

#### **force\_logout\_users**

- Logout all students and tutors except the user who clicks the button.

#### **choose\_lab\_and\_pc**

- A page students are redirected to after registering/logging in to choose their PC number and lab id.
- Procedure:
  - Student chooses a lab from the dropdown.
  - Student chooses a PC number from another dropdown whose options are filtered based on the chosen PC number. (e.g. one lab may have 40 PCs while another may have 50.)

#### **student\_dashboard**

- Displays a list of all the requests sent by the student and their status. Students can make a request from this page.
- Procedure:

## LabLine - Final Delivery

- Display all of the student's requests with their status and queue position.

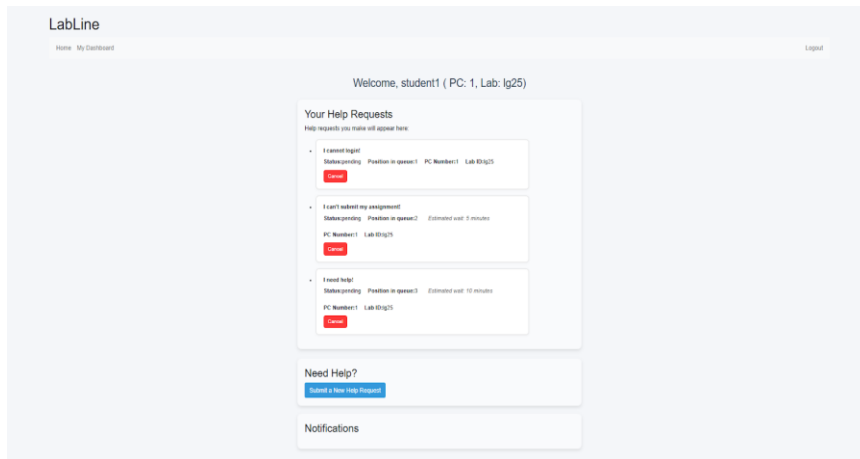


Figure 10. Student Dashboard

### tutor\_dashboard

- Displays a list of all the requests allocated to them and their status. Tutors can update the status of each request to notify the student of their queue position.
- Procedure:
  - Display all of the tutor's assigned requests.
  - Display all of the tutor's pending requests.

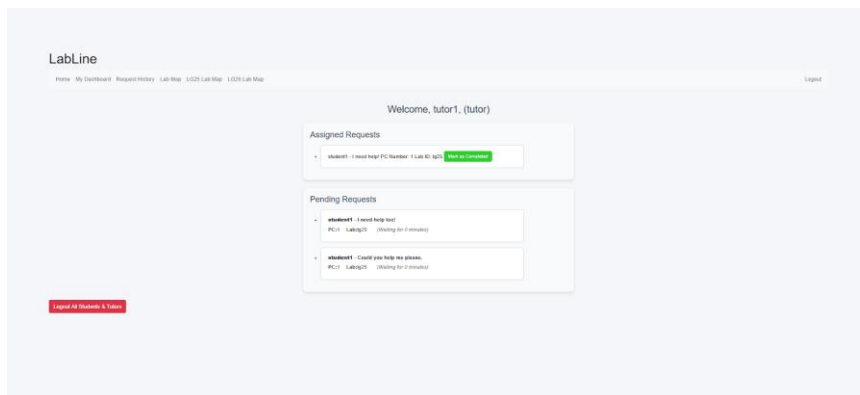


Figure 11. Tutor Dashboard

### lecturer\_dashboard

- Displays all students and tutors on the system, as well as all active requests and completed requests.

## LabLine - Final Delivery

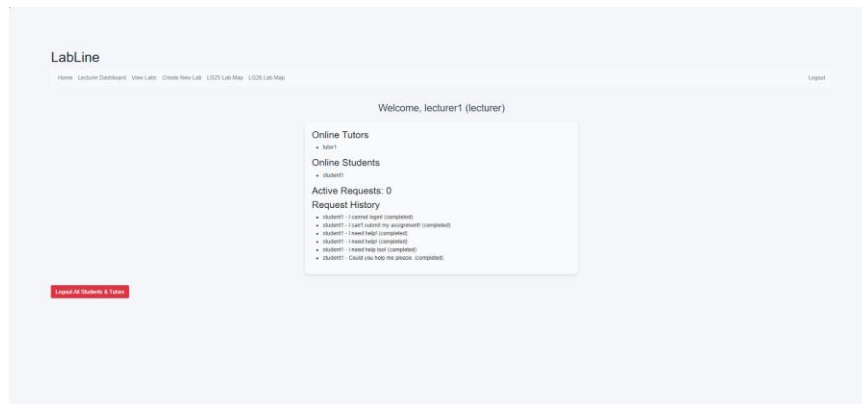


Figure 12. Lecturer Dashboard

### **submit\_request**

- Allows students to send requests for help to a lab tutor. A description field allows students to describe their issue to provide more context to the tutor but is not mandatory.
- Procedure:
  - Students can provide a description of their request and then click submit.
  - The request is then assigned to the next available tutor as part of a round-robin assignment.

### **accept\_request**

- Change the status of the request from “pending” to “in\_progress” and update queue positions.

### **cancel\_request**

- Remove a request sent by a student from their dashboard and from the tutor’s dashboard.
- Procedure:
  - Modify the status of the cancelled request to “cancelled”.
  - Notify the relevant student and tutor of the cancelled request.
  - Notify all students that their position in the queue has been updated due to the cancelled request.

### **request\_history**

- Display a list of a specific lab tutor’s requests, including those which are pending, in progress and completed.

### **create\_lab**

## LabLine - Final Delivery

- Allow lecturers to create their own lab layouts.

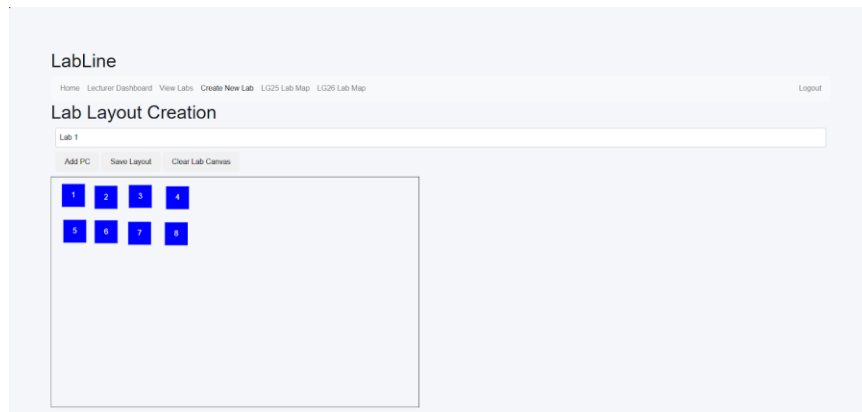


Figure 13. Lab Creation Canvas

### **save\_layout**

- Save the lab layout created by the lecturer to the database.

### **delete\_layout**

- Remove the selected lab layout from the database.

### **view\_layouts**

- List all saved lab layouts created with the create\_lab view.

### **get\_saved\_canvas**

- Load the saved canvas selected in the view\_layouts view

### **Lg25\_map, 26 etc.**

- Example lab layouts where the PCs change from blue to green when a student registers/logs in with that specific PC and specific lab.

## LabLine - Final Delivery



Figure 14. Pre-made Lab Layout

## 3.2 Software Architecture

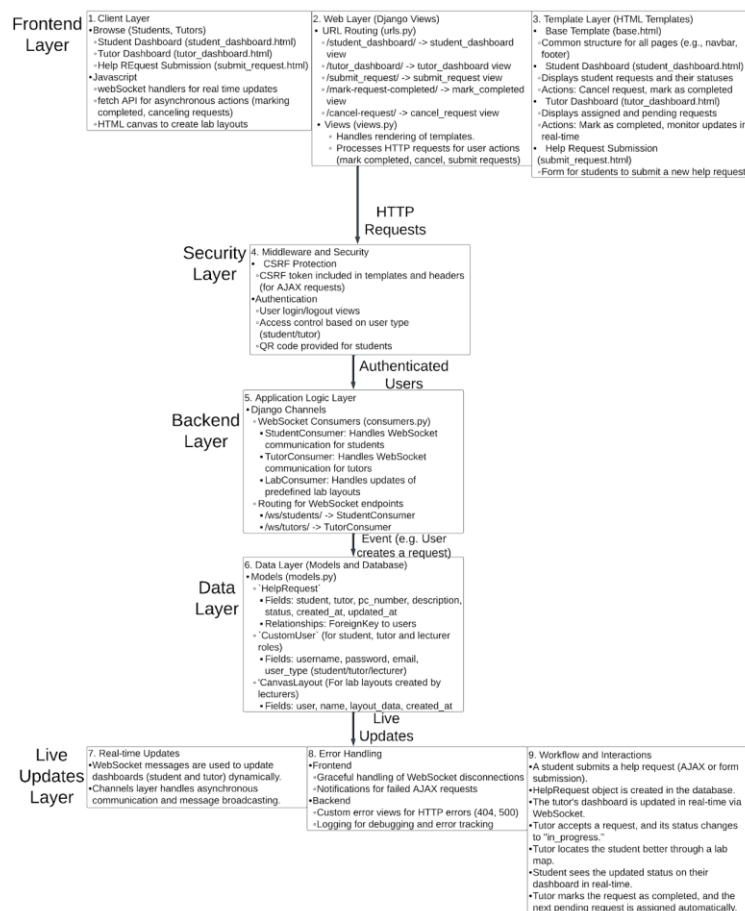


Figure 15. Architecture diagram

The website is included on the front/backend of our system. It includes the UI used by users to navigate through LabLine. It was developed with HTML and CSS. We also optimised the frontend for mobile use.

## LabLine - Final Delivery

The security layer includes CSRF protections where unique tokens are created for each session and each POST request made. This includes students who send help requests for example. Django with Python manages the backend functionality of LabLine. Django manages the main operations of LabLine, including user registration/login, managing requests and lab layouts. Django Channels allow for communication between the user types, including between students and tutors.

A QR code is also included with LabLine to provide a separate registration for those using LabLine to send requests from those using LabLine in more administrative roles such as lab tutors or support agents. This QR code is provided through an API.

The data layer includes the relevant Django models that store information about users and lab layouts.

WebSocket handles the automatic display of requests to lab tutors and status updates to students without the need to refresh their dashboard. WebSocket consumers enable this for students and tutors. This is provided through an Asynchronous Gateway Server Interface or ASGI. This differs from the Django standard WSGI, which is synchronous meaning that it can handle only one request at a time unlike ASGI which can handle multiple.

The data layer includes an SQLite database storing information on users, requests and lab layouts. We have also deployed LabLine to an EC2 instance on AWS, allowing us to conduct user testing with a number of students and tutors.

### 3.3 AWS Diagram

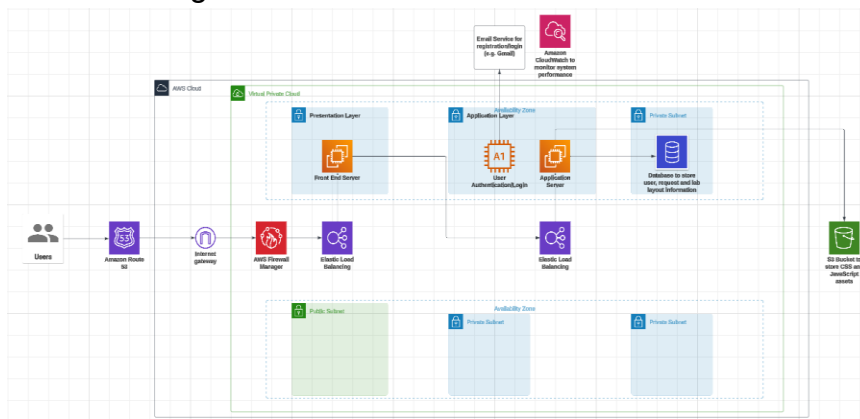


Figure. AWS diagram for fully implemented LabLine

### 3.4 Sitemap Diagrams

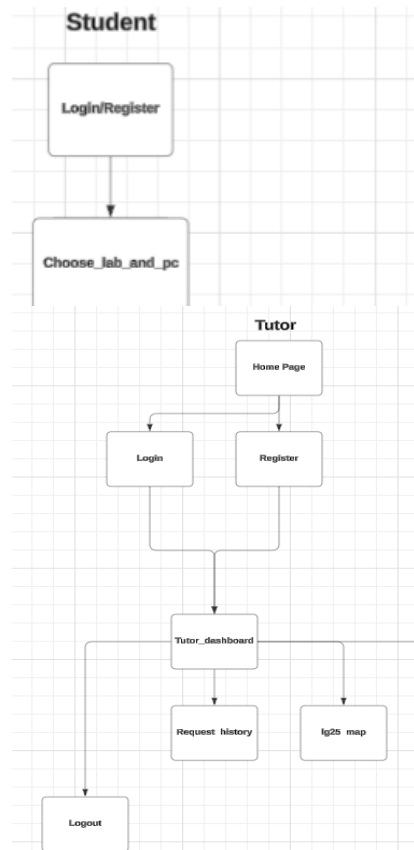


Figure 16. Student Sitemap

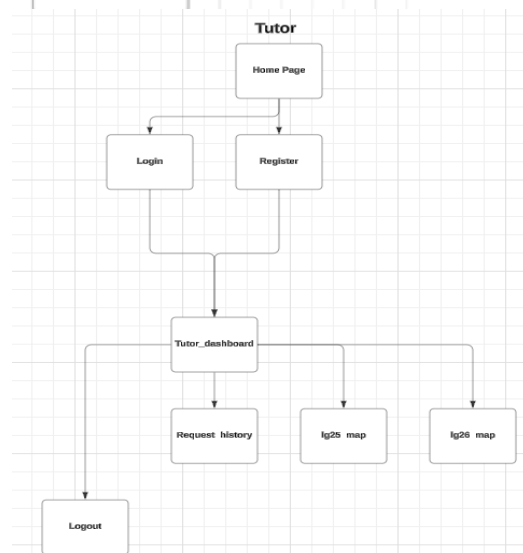


Figure 17. Tutor sitemap

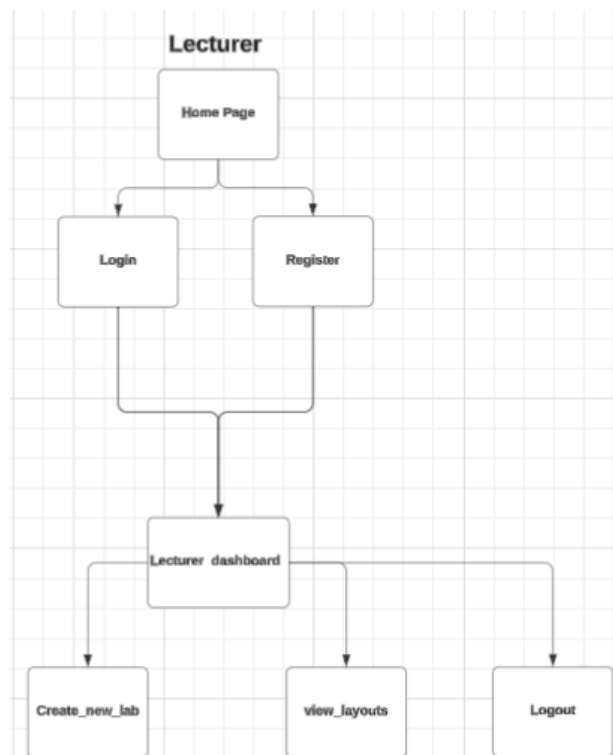


Figure 18. Lecturer sitemap

#### Login and Authentication:

- Student\_register: A registration page exclusively for student users who will access it from a QR code.
- Register: Registration page for tutors and lecturers.
- Login: Login and authentication for users already registered.
- Choose PC and lab: Where students select their PC number and lab so they can be identified by tutors.

#### Dashboards:



## LabLine - Final Delivery

- Student Dashboard: Main page for students to check their sent requests and to receive status updates.
- Tutor Dashboard: Main page for tutors to manage their requests.
- Lecturer Dashboard: Main page for lecturers to manage registered students and tutors.

Request History: Allows tutors to view their previous requests and their statuses.

Send Request: Page where students can provide a description of their issue and send it.

### Labs:

- Lg25 and Lg26 layout: HTML canvas layouts of the lg25 and lg26 computer labs.
- Create a lab: Allows lecturers to create their own lab layouts.
- View labs: View lab layouts that have been created.

## 3.5 Logical Database Design

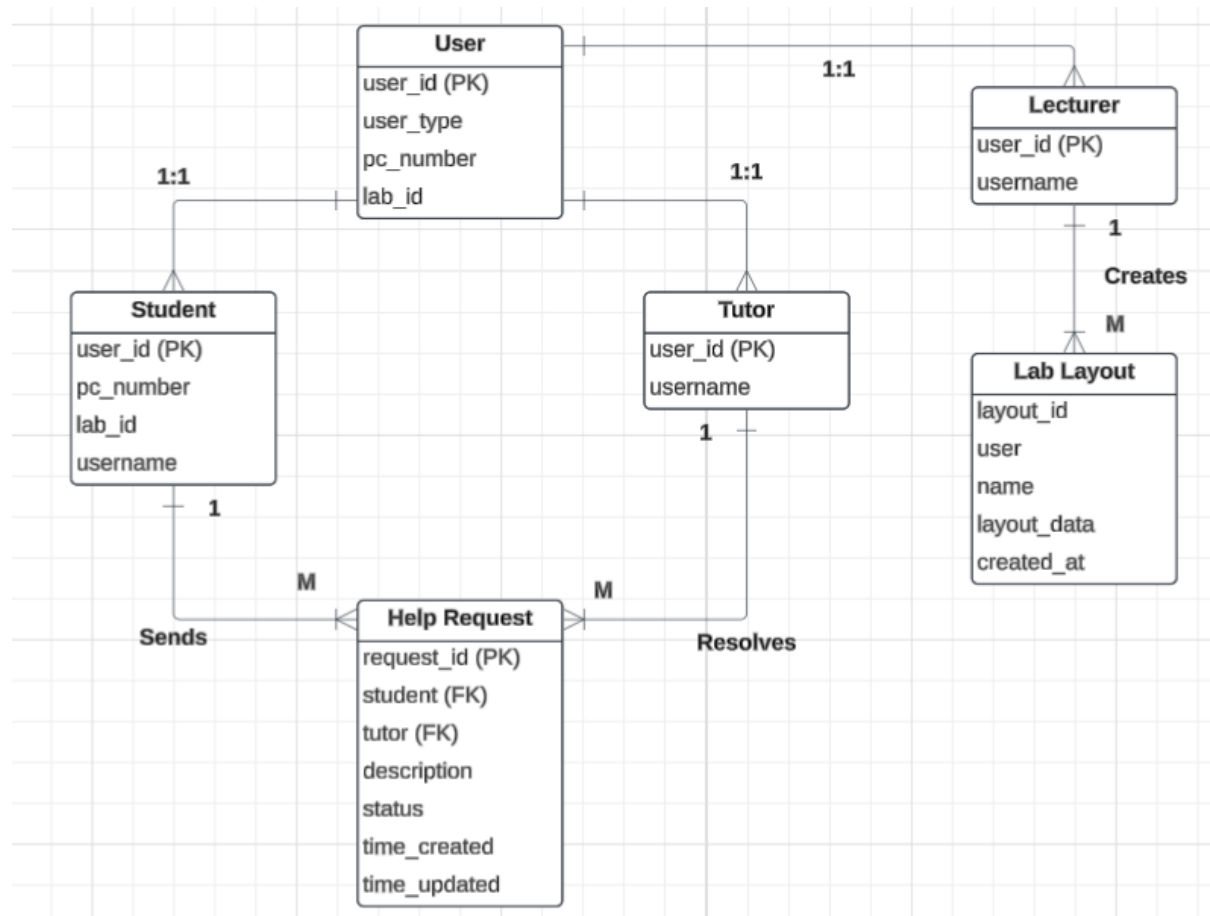


Figure 19. Logical Database Design for LabLine

### Models:

#### **User:**

Represents all users and user types of LabLine. It includes the following attributes:

- User\_id: The primary key for users.
- user\_type: Includes the types of users who use LabLine (students, tutors and lecturers).
- pc\_number: The number of the PC chosen by a student user when they login.
- Lab\_id: The name of the lab/room chosen by a student user.

#### **Help Request:**

Represents help requests sent by students.

## LabLine - Final Delivery

- student: The user who created the request. Linked to the user model.
- tutor: The tutor who accepts the request. Linked to the user model.
- Description: An optional description that students can include with their request.
- status: The progress of the request towards completion. Includes four statuses ('pending', 'in\_progress', 'cancelled' and 'completed')

### Lab:

Represents custom lab layouts created by lecturers:

- Layout\_id: Unique identifier for the layout.
- User: The user who created the layout.
- Name: The name of the layout.
- Layout\_data: The JSON data for the layout.
- Created\_at: Time when the layout was created.
- 

## 3.6 Use Cases

### Management of Requests

The below use case diagram represents the interactions between students and tutors when managing requests.

- Students can view their own dashboard and send requests to tutors for help. They can optionally include a request, a description and cancel/edit the request if either their need changes or they have resolved the request already.
- Tutors also have their own dashboard where they can accept requests and mark them as completed. Students then receive a notification to indicate a change in the status of their request.

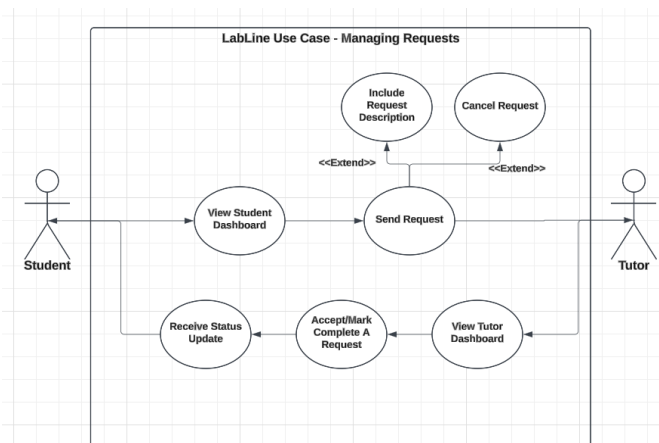


Figure 20. Use case diagram representing the interactions between a student and tutor on LabLine

## User Registration/Login



Figure 21. Use case diagram representing each user type registering for LabLine

## 3.7 Data Flow Diagram



Figure 22. Data flow diagram

**Django:** The server-side framework which helps to manage the flow of data between the user on the left and the database on the right. It receives and processes data from the user, completing tasks with the data and storing or retrieving that data for use.

**SQLite:** The database which contains all information relating to users, requests and lab layouts stored on LabLine. Unlike standard SQL, SQLite is a lightweight database and is built in with every Django web application.

## External Software Interfaces

A unique QR code is provided for students using LabLine to prevent them from registering as tutors or lecturers and potentially disrupting their lab sessions. This QR code is provided in section A5 of the appendix.

## 3.8 Performance Requirements

The following are some of the key performance requirements of our website, Labline which ensure that it operates efficiently:

### Responsiveness

## LabLine - Final Delivery

Pages should load within 2 seconds, User actions(e.g. submitting/cancelling a help request) must reflect changes in under 1 second. Websocket based updates should reach all connected clients within 300 milliseconds of a server side event.

### Real time communication

The websocket connections must remain stable throughout lab sessions. The system should support a large number of concurrent websocket connections without delays or dropped messages. The updates(e.g. The queue positions, tutor assignments) must be synchronised across all of the relevant users in real time.

### Concurrency

Labline is expected to handle a large number of simultaneous users without affecting the responsiveness of the website, tutor and student dashboards must independently reflect live updates, regardless of the user load.

### Security

CSRF protection is enforced on all forms. The user authentication and role-based authorisation guard the access to dashboards and the data which is not shared with students. Only authorised users can, "Accept" or "mark request as completed"(tutors), or View or manage lab usage(lecturers). Sensitive actions such as forced user log out, and lab creation requires POST and CSRF tokens with confirmation. HTTPS should be enforced in production to protect all data transit.

### Scalability

Designed to support additional labs, users, and roles without major refactoring. The AWS infrastructure should support scaling up and down based on demand by monitoring the performance. The system should also be deployable across multiple campuses with just minimal configuration changes.

## 3.9 Working Prototype Software (Video Link)

Link to prototype video -  
<https://drive.google.com/drive/folders/1eE9KiKoHzhRMtuqNzKhkiZ84QlyLssks?usp=sharing>

## 3.10 Description of Technical Challenges Solved Bringing the Idea to Market

Below are the technical challenges that we needed to solve to bring Labline to market:

### Real time communication between users using django channels

Django is a synchronous web framework by default, however, implementing live updates for help requests across the different user types (student, tutors, and lecturers) required asynchronous support.

## LabLine - Final Delivery

To solve this problem, we configured the project to use Django channels and ASGI, which then enabled us to use websocket consumer. We created DashboardConsumer to handle updates for users using `user_{id}` groups. To handle the lab wide updates, for example the updates to the lab maps, we created the LabConsumer. To push the updates made by users in real time we created role specific broadcast logic using `async_t_sync(channel_layer.group_send)`. This architecture allowed us to deliver fast and targeted websocket updates without the need to manually refresh the webpages.

### Queue management and Fair tutor assignment

We need to fairly assign the help requests made by students to tutors so that the help requests were shared between the logged in tutors and tutors were not overwhelmed. We also need to ensure that students received live feedback on their position in the queue and the estimated wait time for each of their requests so they know how long they would be waiting before receiving help.

The solution we used for this challenge was implementing a round robin tutor assignment system. This means that the first request created is assigned to the first logged in/online tutor, each online tutor is assigned a request before the first tutor is assigned a second request. This approach simplified the load balancing for tutors. When a student submits a help request their ID is rotated to the back of the queue, the queue position and estimated wait time are recalculated and sent to students using websocket notifications.

### Role based dashboards and views

Labline needed to serve different users each with different views, permissions and actions, all under a single codebase and user model.

Our solution to this was using Django's custom CustomUser model and adding a `user_type` field to differentiate the roles. The views and templates are gated based on this value, and we created role specific “dashboards” (student dashboard, tutor dashboard, and lecturer dashboard).

- The student dashboard displays personal help requests created by the student, their position in the queue, and real time updates and notifications.
- The tutor dashboard shows pending and in progress requests, with dynamic control over which request can be accepted (discussed in more detail below)
- The lecturer dashboard displays the system wide metrics and user control options, including forced logouts.

By using this method, we were able to ensure the different users only see the relevant functionality based on their `user_type`.

### Single accept button logic for tutors

To make the system more user friendly and accurate (by reducing chances of mis clicking and accepting a request which isn't next in the queue) for tutors, only the first request from the list of pending requests should have the accept button, and once accepted it moves to in\_progress status and the next request in the queue becomes active.

To address this challenge, we used JavaScript in the tutor dashboard file to hide all the accept buttons except for the first request. Once that request is marked as completed and the assigned requests list is empty the next request in the pending requests list automatically gets the accept button. This prevents double assignment bugs during high activity labs and ensures the system runs smoothly for tutors.

### Live lab maps with colour coded pc squares

To allow tutors to better locate students, a lab map can quickly provide the location of the student providing the request. The student chooses the lab and PC number closest to them and the corresponding PC on the lab map changes from blue to green. When the student sends a request and while they have requests that are still pending, their PC on the lab map changes from green to red.

To implement this, I had to create an api fetch between the pc-data view function and the template of the corresponding lab. For example, if loading the lg25 map, the pc\_data function gets the corresponding lab id and filters the students who have chosen lg25 when logging in and if they have any pending requests. This is then appended to the JSON data and provided to the lg25 template through an API call.

The data is then filtered into students who have logged in but do not have pending requests and those that do.

### Secure role based access and session management

The challenge we faced was that sensitive actions like logging all students and tutors out or seeing help requests raised by students, needed to be secure and limited to the correct user types (tutors, and lecturers).

To ensure security and confidentiality we protected all the sensitive views with the @login\_required decorator. We also made sure that students could not register as lecturers or tutors and get access to restricted information and features, we accomplished this by created a new register view specifically for students(studen\_register) in which we made it so the user\_type 'student' was forced, meaning no matter what user type is selected from the drop down while

registering, the user would still be registered as a student. We also created a QR code for students to scan to register, which directed them to this registration form.

### AWS deployment

We needed to conduct user testing for LabLine with multiple users and required a way for them to access it from their own devices.

We decided to create an EC2 instance on AWS to deploy LabLine. We created a URL that can be accessed by tutors and lecturers while the instance is running.

## 3.11 Source Code Highlights

### **Additional Snippets:**

<https://drive.google.com/drive/folders/1anAev5od2XQzVzrxwh50O92Yq64qS93H?usp=sharing>

### User Registration

```
def register(request):
    if request.method == 'POST':
        form = CustomUserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)

            # Redirect users based on user type
            if user.user_type == 'student':
                return redirect('choose_lab_and_pc')
            elif user.user_type == 'tutor':
                return redirect('tutor_dashboard')
            elif user.user_type == 'lecturer':
                return redirect('lecturer_dashboard') # New dashboard for lecturers
        else:
            form = CustomUserCreationForm()
    return render(request, 'register.html', {'form': form})
```

Figure 23. Register view function

This code handles the registration and authentication of new users using LabLine. It provides a form to the user to enter their details and checks if the user's details are valid. If they are valid, it saves the user's details, logs them in and redirects them to the appropriate pages based on their user type.

### **Code Explanation:**

- A form “**CustomUserCreationForm**” is provided for the user.
- With “**form.is\_valid()**”, it checks if the user's entered details are correct and saves if so, “**form.save()**”.
- Followed by if statements to determine whether the new user is a student, tutor or lecturer and redirects them.

### Choosing a PC number and Lab ID



## LabLine - Final Delivery

```
def choose_lab_and_pc(request):
    lab_pc_count = { # Indicate the number of PCs available to choose from each lab.
        'lg25': 50,
        'lg26': 50,
    }

    if request.method == "POST":
        form = PCNumberForm(request.POST)
        if form.is_valid():
            pc_number = form.cleaned_data["pc_number"] # Extract PC number from form.
            lab_id = form.cleaned_data["lab_id"]
            request.user.pc_number = pc_number # Assign to logged-in user.
            request.user.lab_id = lab_id
            request.user.save() # Save user model.
            messages.success(request, f"Your PC number is {pc_number} in lab {lab_id}.")
            return redirect("student_dashboard")
        else:
            lab_id = request.GET.get('lab_id', None)
            form = PCNumberForm()

            if lab_id in lab_pc_count:
                available_pcs = [(i, str(i)) for i in range(1, lab_pc_count[lab_id] + 1)] # List the available
                form.fields["pc_number"].choices = available_pcs
                form.fields["lab_id"].initial = lab_id

    return render(request, "choose_lab_and_pc.html", {"form": form})
```

Figure 24. Choose lab and PC view

This code provides options for the student to choose their lab location and their closest PC. It extracts the details entered by the student and assigns the lab id and PC to the student upon entering them. It then redirects the student to their dashboard.

## Student Dashboard

```
#login_required
def student_dashboard(request):
    user = request.user
    help_requests = HelpRequest.objects.filter(student=user)

    queue = list(
        HelpRequest.objects.filter(status="pending").order_by("created_at")
    )

    AVERAGE_WAIT_PER_REQUEST = 5 # minutes
    request_positions = {
        req.id: idx + 1 for idx, req in enumerate(queue)
    }

    for req in help_requests:
        req.queue_position = request_positions.get(req.id)
        if req.status == "pending" and req.queue_position:
            req.estimated_wait_time = (req.queue_position - 1) * AVERAGE_WAIT_PER_REQUEST
        else:
            req.estimated_wait_time = None

    return render(request, "student_dashboard.html", {"help_requests": help_requests})
```

Figure 25. Student dashboard view

This code handles the student's dashboard including their created help requests, an ordered queue of the student's help requests and the student's estimated wait time.

## Tutor Dashboard

## LabLine - Final Delivery

```
@login_required
def tutor_dashboard(request):
    user = request.user
    assigned_requests = HelpRequest.objects.filter(tutor=user, status='in_progress')
    pending_requests = HelpRequest.objects.filter(status='pending', tutor=user)

    for req in pending_requests:
        if req.created_at:
            delta = datetime.now(timedelta.utcnow()) - req.created_at
            req.waiting_minutes = int(delta.total_seconds() // 60)
        else:
            req.waiting_minutes = 0

    return render(request, 'tutor_dashboard.html', {
        'assigned_requests': assigned_requests,
        'pending_requests': pending_requests,
    })
```

Figure 26. Tutor dashboard view

This code handles the tutor's dashboard. It lists their assigned and pending requests and how long the student associated with each pending request has been waiting for it to be accepted.

### Submitting Requests

```
def submit_request(request):
    if request.method == 'POST':
        form = HelpRequestForm(request.POST)
        if form.is_valid():
            help_request = form.save(commit=False)
            help_request.student = request.user
            if ONLINE_TUTORS:
                selected_tutor_id = ONLINE_TUTORS.popleft()
                ONLINE_TUTORS.append(selected_tutor_id)
                selected_tutor = CustomUser.objects.get(id=selected_tutor_id)
                help_request.tutor = selected_tutor
            help_request.save()

            # Notify tutors
            channel_layer = get_channel_layer()
            if help_request.tutor:
                from datetime import datetime, timedelta
                delta = datetime.now(timedelta.utcnow()) - help_request.created_at
                waiting_minutes = int(delta.total_seconds() // 60)

                async_to_sync(channel_layer.group_send)(
                    "tutor_{}".format(help_request.tutor.id),
                    {
                        "type": "update_dashboard",
                        "message": ("New help request from {request.user.username}: {help_request.description}",
                                "event_type": "new_request",
                                "request_id": help_request.id,
                                "description": help_request.description,
```

```
channel_layer = get_channel_layer()

# Calculate queue position and estimated wait time
pending_queue = HelpRequest.objects.filter(status="pending").order_by("created_at")
queue_position = list(pending_queue).index(help_request) + 1
estimated_wait = (queue_position - 1) * 5

# Notify the student with full details
notify_dashboard(
    user_id=help_request.student.id,
    message="Your request '{help_request.description}' has been received.",
    event_type="status_update",
    request_id=help_request.id,
    new_status="pending",
    description=help_request.description,
    student=help_request.student.username,
)

return redirect('student_dashboard')
else:
    form = HelpRequestForm()
    return render(request, 'submit_request.html', {'form': form})
```

Figure 27. Submitting requests view

Figure 28. Submitting requests view

This code handles sending student requests to tutors on their dashboard. It checks for online tutors and moves the first tutor available in the queue from the front to the back and associates them with the sent request.

It then notifies the selected tutor with the details of the request such as the estimated wait time and description asynchronously with Django channels.

Then every request in the pending queue is moved up one position and the associated student is notified that their request has been sent and redirected to their dashboard.

### Tutors Accepting Requests

## LabLine - Final Delivery

```
def accept_request(request, pk):
    help_request = get_object_or_404(HelpRequest, pk=pk, status='pending')

    # Make sure the tutor is only accepting their own assigned request
    if help_request.tutor != request.user:
        return redirect('tutor_dashboard')

    # Mark request as in progress
    help_request.status = 'in_progress'
    help_request.save()

    # Notify the student
    notify_dashboard(
        help_request.student.id,
        f"Your request '{help_request.description}' has been accepted by {request.user.username}.",
        event_type="status_update",
        request_id=help_request.id,
        new_status="in_progress",
        student=help_request.student.username,
        description=help_request.description
    )
)
```

Figure 30. Accept request view

```
# Notify the assigned tutor (the one accepting it) to update their dashboard
channel_layer = get_channel_layer()
if help_request.tutor:
    async_to_sync(channel_layer.group_send)(
        f"tutor_{help_request.tutor.id}",
        {
            "type": "update_dashboard",
            "message": f"The request '{help_request.description}' was accepted.",
            "event_type": "status_update",
            "request_id": help_request.id,
            "new_status": "in_progress",
            "student": help_request.student.username,
            "description": help_request.description,
            "pc_number": help_request.student.pc_number,
            "lab_id": help_request.student.lab_id,
        }
    )

broadcast_queue_positions()

return redirect('tutor_dashboard')
```

Figure 29. Accept request view

This code handles tutors accepting a student's request. When a tutor accepts a request, the request's status changes to In Progress. The student is then notified through Channels that their request has been accepted and the queue positions are updated to reflect this.

## Saving Created Lab Layouts

```
def save_layout(request):
    if request.method == 'POST':
        try:
            # Get data from the frontend
            data = json.loads(request.body)
            PCs = data.get('PCs', [])
            lab_name = data.get('name', 'Unnamed Lab')

            for i, PC in enumerate(PCs):
                if 'id' not in PC:
                    PC['id'] = i + 1

            # The user is authenticated and logged in.
            user = request.user

            # Save the lab layout to the database
            layout = CanvasLayout(user=user, name=lab_name, layout_data=json.dumps(PCs))
            layout.save()

            return JsonResponse({'success': True})
        except Exception as e:
            return JsonResponse({'success': False, 'error': str(e)})
    return JsonResponse({'success': False, 'error': 'Invalid request'})
```

Figure 31. Save Lab Layout view

This code handles saving lab layouts created by lecturers. It parses the JSON formatted content in the layout. It then gets the list of PCs from the JSON and the lab name. Unique IDs are then assigned to each PC created in the layout.

It checks if the user is authenticated and saves the layout as an instance in the CanvasLayout model.

## Asynchronous Updates with Django Channels and WebSocket

```
async def update_dashboard(self, event):
    message = {
        "message": event.get("message"),
        "type": event.get("event_type"),
        "request_id": event.get("request_id"),
        "new_status": event.get("new_status"),
        "description": event.get("description"),
        "student": event.get("student"),
        "queue_position": event.get("queue_position"),
        "pc_number": event.get("pc_number"),
        "lab_id": event.get("lab_id"),
        "estimated_wait_time": event.get("estimated_wait_time"),
        "waiting_minutes": event.get("waiting_minutes"),
    }
    await self.send(text_data=json.dumps(message))
```

Figure 32. Asynchronous message sending between user types.

This code notifies students and tutors with status updates of requests using WebSocket that allows users to view updates without the requirement to refresh the page. The content is included in a message variable that is sent in JSON format.

### 3.12 Interface Rationale

The interface design for LabLine prioritises clarity, accessibility and responsiveness for all the users (students, tutors, lecturers). Each page is structured with a minimalistic layout and the overall aesthetic uses a clean style, with light backgrounds and dark text. This style reduces visual noise and cognitive noise to make the application more usable for everyone, especially neurodivergent people, who cannot get accustomed to different bright colours and other changes easily. The navigation bar is tailored based on the user types so it only provides the links each user type will need to use the system as opposed to providing all links for everyone.

#### Navigation and Styling choices

As mentioned above the navigation bar is role specific so based on the users, user type (student, tutor, lecturer) it provides different links and guides the users directly to the relevant actions and content. The navbar is also collapsible which enables the user to use the website on their phone or other device with a smaller screen without any problems.

The soft borders, generous padding and subtle shadows keep content visually organized.

#### Student Interface

The Student Dashboard provides the real time view of help requests created by the student, current statuses of each help request and queue positions.

Cancel and submit buttons are clearly styled with bright colours and easy to access.

Notifications appear live through the websocket set up, which eliminates the need to manually refresh the page.

Students often need quick, reassuring feedback on their requests status and position in the queue. The simple layout with status tags and a clearly labeled button “submit a new help request”

## LabLine - Final Delivery

supports that goal. The feedback we received from our user testing with over 20 students supports our UI design approach and justifies the design of our website.

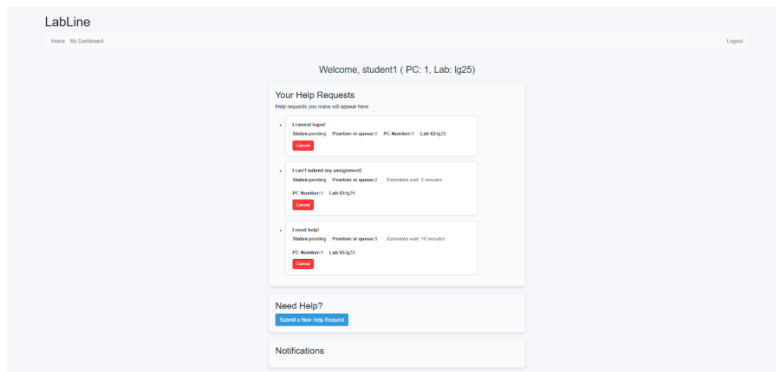


Figure 33. Student dashboard

How easy was LabLine's UI (User Interface) for you to understand and navigate?

5 responses

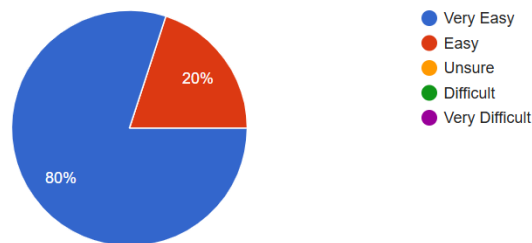


Figure 34. UI survey results

### Tutor Interface

The tutor dashboard offers two panels: Pending requests and Assigned requests, the pending requests panel is populated by the help requests raised by students and only the first request in the list gets a 'Accept' button next to it. Once the request is accepted, it moves to the assigned request panel, and once it is completed, it is removed from the assigned requests panel, and the next request in the pending request list/panel gets the 'Accept' button. So at any given point, only one 'accept' button is present to simplify the dashboard and request management. All updates are pushed in real time for both students and tutors via websockets, and the requests are visually separated for more clarity. The Tutor dashboard also features a "log out all students and tutors" button at the bottom, which the tutors can use to log out all the students at the end of lab sessions. This button is in a bright red casing, ensuring users are made aware of it, and it is not used unless tutors are instructed to use it by lecturers at the end of lab sessions.

## LabLine - Final Delivery

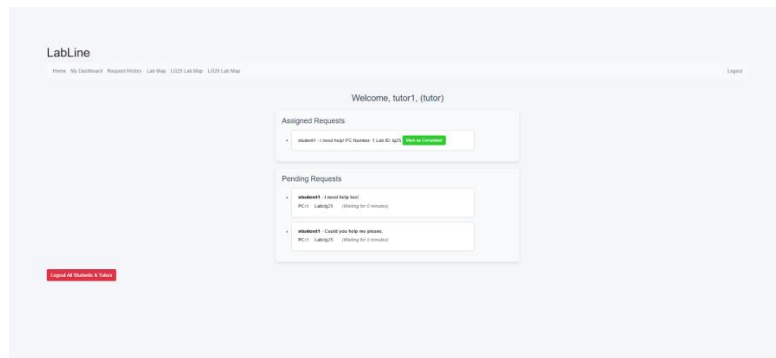


Figure 35. Tutor Dashboard

### Lecturer Dashboard

The Lecturer Dashboard features the information of all logged in tutors and students, showing the list of active requests and request history, the Lecturers also get the “log out all students and tutors” button at the bottom of their dashboard with which they can log all users out at the end of lab sessions. Lecturers need visibility and not interaction heavy controls on their dashboard. The layout gives them insights without cluttering the UI with unnecessary buttons or details.

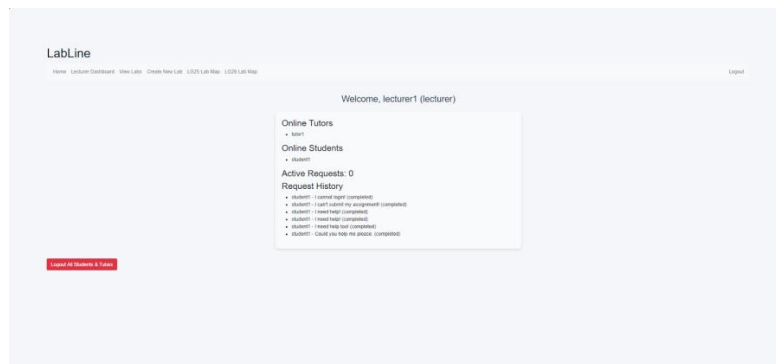


Figure 36. Lecturer dashboard

## Appendix

### A1. Team CVs

#### Eoin Francis

Accomplished final year BSc Computing for Business student with a genuine passion for software development and IT. My academic journey has equipped me with a solid foundation in computer science principles. Eager to apply knowledge and further develop skills. Positive reputation for punctuality, determination and ability to apply informative feedback.

### Strengths:

I have gained experience in working in teams throughout my four years in DCU on a variety of projects from software development to report writing.

I also believe that my public speaking skills have improved through group presentations and also interactive orals. In particular, I was initially nervous when presenting in front of a whole class but have learned that a good level of preparation can reduce the stress.

### Experience:

During the past four years in DCU, I have developed skills and gained experience across several areas of enterprise computing, including web design/development, database management and machine learning through assignments and projects. I developed a combination of technical skills through tools such as HTML, Python, SQL and Django and soft skills such as project and team management and presentation skills.

### Skills:

#### **Technical:**

- HTML, CSS and Bootstrap.
- Python.
- Django.
- JavaScript.
- MySQL.

#### **Soft Skills:**

- Attention to detail.
- Project management.
- Teamwork.

#### **Project Role:**

- Secondary market research for the project.
- Added ability for lecturer users to create and save lab layouts.
- Deployment of the project through an AWS EC2 instance.

#### **Addressed Skills Gaps:**

- Became more comfortable with JavaScript syntax. Previously used JavaScript only for simple events such as displaying text when clicking a button.

## LabLine - Final Delivery

- Gained better experience in agile/scrum software development through regular meetings as a pair to discuss progress over a period of months. This was something I missed out on due to not securing an INTRA placement.

### Weaknesses:

While I have good participation in group settings, I still have a tendency to take on leadership roles and can sometimes unnecessarily push group members to meet demands they might be able to achieve. I believe the paired nature of this project meant that both of us took on notable roles during the project.

## **Harsh Saini**

Motivated final year Computing for Business student with practical experience in full stack web development. Over the course of my academic journey I was able to build and develop new skills and knowledge in computing and business principles.

### Strengths

Over the course of my degree I have worked extensively in both solo and group projects. My internship and INTRA placement at Bank of Ireland's Cloud Centre of Excellence gave me industry exposure and I have developed a strong work ethic and strong problem solving skills.

### Experience

I completed an 8 month INTRA placement and a summer internship at Bank of Ireland's Cloud Centre of Excellence, some of the tasks included:

- Worked in the Cloud Business Office under the supervision of the Cloud Product Lead with a focus on the management of projects.
- Supported the automation of routine tasks to improve operational efficiency.
- Participated in governance meetings to track project progress and ensure timely delivery.

### Skills

#### **Technical:**

- HTML, CSS
- Python
- Django
- Javascript(Basic-Intermediate)



**Soft skills:**

- Problem solving
- Communication
- Project management
- adaptability

**Project roles:**

- Real time websocket communication logic
- Queue management and fair tutor assignment
- Student - tutor dashboards
- Javascript updates for live dashboard changes

**Addressed skill gaps:**

This project helped me become more confident using Django, and I also improved my JavaScript proficiency and now i feel more capable of integrating frontend/backend functionality. It taught me to work through the errors and seek feedback regularly

Weaknesses

I struggle with public speaking and experience social anxiety in group settings. I'm actively working on improving these skills through practice and feedback. Time management has also been a challenge in the past, but through structured planning and help from my partner I have seen some improvement.

## A.2 Business Model Canvas

### The Business Model Canvas

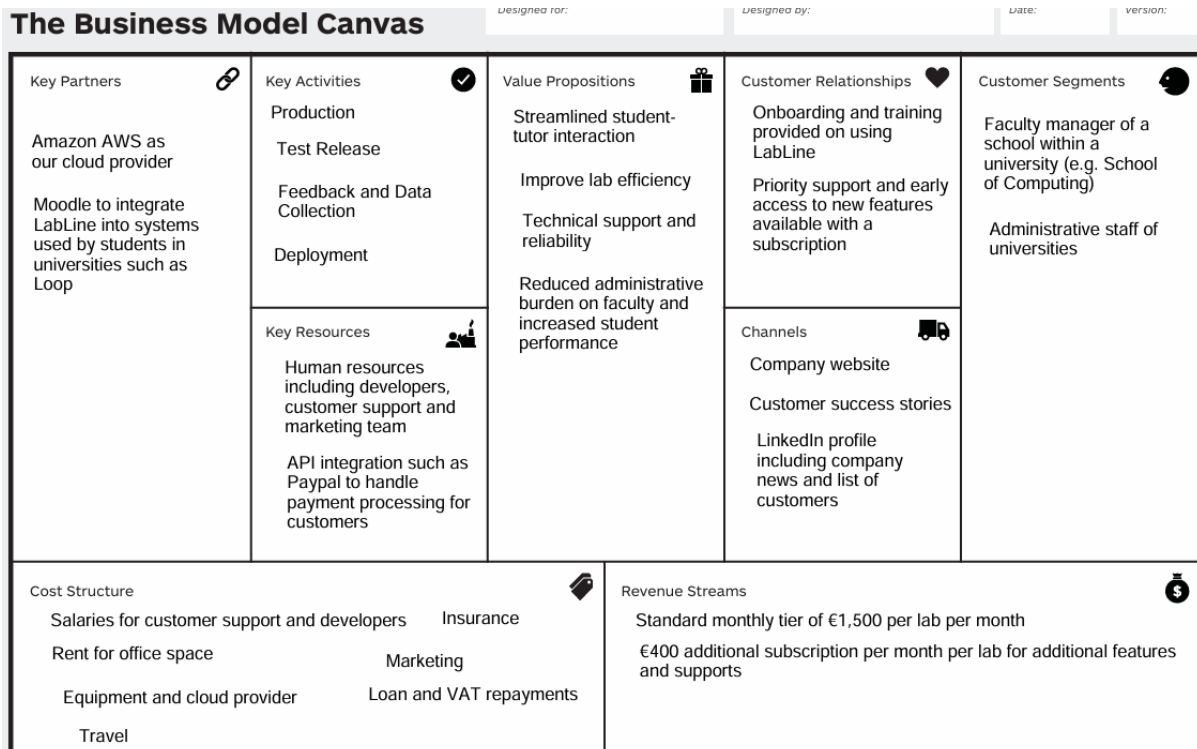


Figure 37. Business Model Canvas

## A.3 Value Proposition Canvas

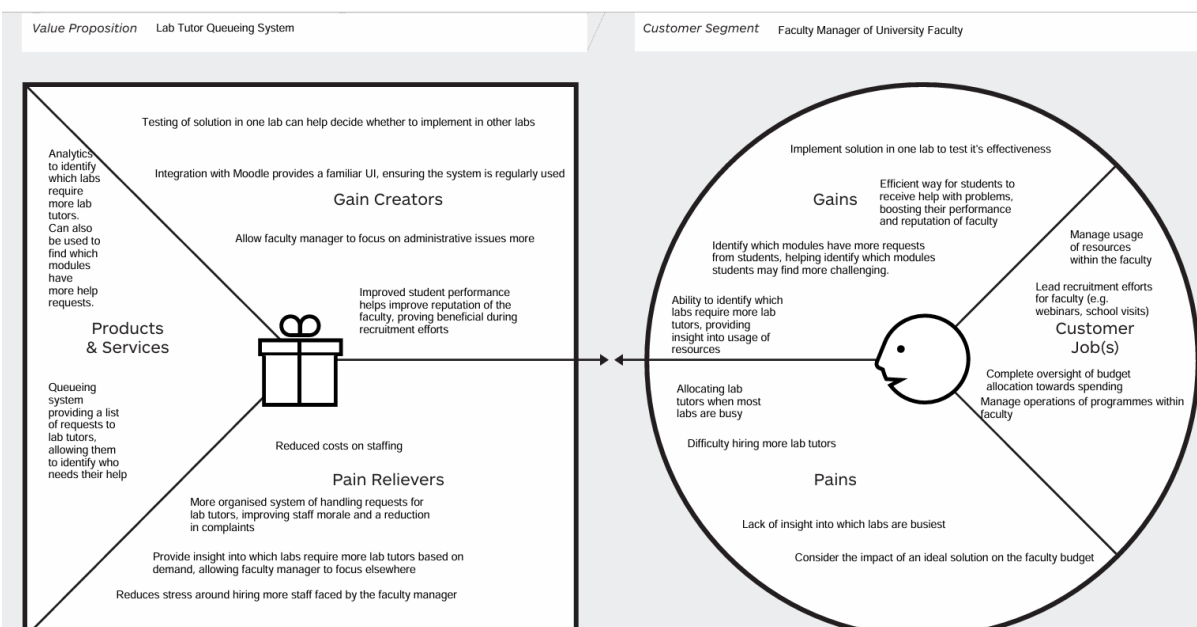


Figure 38. Value Proposition Canvas

## A5. LabLine Instructions For Students



Figure 39. QR code

### Signing Up:

Scan the QR code above

At the bottom of the page under “New Here?” click “Register”.

Enter your username under the “Username” text box.

Enter your password for the new user. Re-enter the password for verification purposes. Do not use the password for your DCU account or any other account that you have.

Once the above details are entered, click “Register” to register as a new user.

After signing in, you will be asked for a “PC Number” and a “Lab ID”.

PC Number represents the number allocated to each PC in the computer labs. It can be found on a white sticker usually stuck onto a PC. You do not need to use a lab PC to use Labline, but indicating the PC number closest to you helps lab tutors locate you easier.

Lab ID represents the room where you are located. In the School of Computing, this is usually in the format LXYY, where X represents the floor the room is located and YY represents the number of the room. (e.g. L128 means room number 28 located on the first floor.)

Once these details are entered, you will be redirected to the Student Dashboard.

### Student Dashboard (Overview):

The Student Dashboard is where students can send help requests to lab tutors and view the statuses of their previous requests.

Upon loading the Student Dashboard for the first time, the “Your Help Requests” section, which includes all of the requests you have sent, will be empty.

### Send a Help Request:

To create a new help request, click the “Submit a new help request” under the “Need Help” section. You will be redirected to the Submit Help Request page. There is an option to add a description of your request on the page, but this is optional. Click “Submit” to send a help request.

Student Dashboard (Managing Your Requests):

Once you have clicked “Submit”, you will be redirected to the Student Dashboard with your request now appearing under the “Your Help Requests” section.

Your request also includes its position in the queue. A “Cancel” button is included if you resolve the problem you are facing before a lab tutor has arrived.

Request Status:

Initially, the status of your request is set to “pending”, meaning that a tutor has yet to accept your request.

When a tutor accepts your request, the status of your request will update to “in\_progress”.

When a tutor has resolved your request, the status of your request will update to “Done” and the cancel button will be removed.

Student Dashboard (Notifications)

As the status of your request is updated, you will receive notifications under the “Notifications” section.

Logging Out and Logging In:

Once you are finished using Labline, click the “Logout” option, you will be redirected to the Login page.

If you decide to log back into Labline after logging out, enter your username and password under “Login” and click Login.

When you log back in, you will be asked again for your PC number and Lab ID. This is because it is possible for some students to switch locations during a lab.

**A6. LabLine Instructions For Lab Tutors**

Url for Labline: <http://13.60.173.241:8000/>

Instructions:

Signing Up:

Click Login/Register to sign up as a user.

At the bottom of the page under “New Here?” click “Register”.

Enter your username under the “Username” text box.

Enter your password for the new user. Re-enter the password for verification purposes. Do not use the password for your DCU account or any other account that you have.

Once the above details are entered, click “Register” to register as a new user.

## LabLine - Final Delivery

Once these details are entered, you will be redirected to the Tutor Dashboard.

### Tutor Dashboard (Overview):

The tutor dashboard is where tutors can manage help requests sent by students.

The “Pending Requests” section includes a list of requests sent by students. Only the first request in this section has an “Accept” button.

Refreshing the page will allow you to see the PC number and Lab ID associated with a request. Note that this does not necessarily mean that the student will be using that specific PC number, but will help to locate the student.

Clicking the accept button moves the request to the “Assigned Requests” section. Clicking this means that you are now ready to resolve the request associated with it, since the Accept button only appears if there are no requests within the Assigned Requests section.

Once you have resolved a request, click “Mark as Completed” to remove it from your Assigned Requests section, allowing you to accept the next request in your Pending Request section.

### Previous Requests:

Clicking on “Previous Requests” will allow you to view a history of the requests that have appeared in Pending Requests. The details of each request include the user who sent them, a description of the request if added by the user, as well as when the request was initially sent and when it's status was most recently updated.

## A.7 Financial Overview

### Angel Investment

We were able to secure an investment of €50,000, from friends and family that supported, and believed in our business, and were confident they would receive a return on their investment once we bring this idea to market.

### Founder investment

Eoin and Harsh will invest 25,000 each from their personal savings to help with the initial startup costs of the business.

### Operating Revenue

We aim to onboard universities from February of year 1, with a trial period of 3 months on average for each university to measure the performance and impact of using Labline before fully committing to the monthly contracts.

### **Investment income:**

Angel investment of €50,000 received in January from friends and family who believe in our business and are confident they will receive a greater return on their investment. Founder investments of €25,000 made by Harsh and Eoin from their personal savings in January, this will greatly support the business in the first year with all the costs of operation. We will be taking out a business loan of €40,000 from AIB in June for a term of 36 months to support the business. The total capital raised for the first year is €140,000.

### **Operational Revenue:**

The business is projected to onboard 10 labs by the end of Year 1, and have 6/10 labs signed up for the additional subscription plan. Each lab pays a €1,500 monthly subscription fee (as mentioned in the pricing model), and 6 institutions also pay for the premium plan which costs €400 per lab per month. This brings the total operational revenue generated to €121,400. We gradually increase the number of labs using our product through careful marketing strategies and by attending educational institutions around Ireland to demonstrate the capabilities, and advantages of implementing Labline. In year 2 we are projecting to grow by 11 labs, so in total we will have 21 labs onboarding by the end of year 2, along with an additional 6 labs on the premium plan bringing the total to 12 labs on the premium plan at the end of year 2 for a total operational revenue of €327,900. Year 3 kicks off with the same number of labs as the end of year 2 at 21 labs, for the first quarter (Jan - Mar). We begin to onboard an additional 4 labs in the second quarter at the start of April along with an additional 2 more labs on the premium plan. Our total lab count for the end of year 3 is 25 labs on the basic plan with 14 of these labs also paying for the premium plan. After the second quarter of year 3 our focus shifts from onboarding new clients to maintaining and assisting the current customers, and maintaining the system before we progress further and expand to more customers. The total operating revenue for year 3 is €496,800. There is slow growth in the first 2 quarters of each year since the educational institutes are opted in for a trial of Labline which is usually trialled on 2 labs with 1 premium plan, before they commit to fully onboarding it to multiple labs or keep it on the existing ones.

### **Bank loan**

As mentioned in the investment Income above we will be taking out a business loan from AIB of €40,000 in the month of June of year 1 for a period of 36 months. The loan comes with an interest rate of 7.45% which makes the monthly repayments to be €1,245.47 for 36 months. This makes our total amount repayable to €44,836.96. This loan is extremely important for us in the first year of our business since it will make it financially possible to hire employees, and fund other aspect of our business. [23]

|  |                                  |                             |   |
|--|----------------------------------|-----------------------------|---|
| Monthly repayments<br><b>€1,245.47</b> | Interest rate<br>7.45 %          | Total fees<br>€0            | Loan amount<br>€40,000                  |
| Loan length<br>36 months               | Representative APR<br>7.82 % APR | Cost of credit<br>€4,836.96 | Total amount<br>repayable<br>€44,836.96 |

Figure 40. Bank Loan amount

### Monthly Expenses

salaries for employees:

Initially we will be employing 2 developers, 2 customer support and success staff, and 1 part marketing staff for the first year. We will be paying €3,500 per month to developers, €2,000 per month to each customer support staff and €1,200 to the part time marketing staff.

We are aiming to hire an additional developer and customer support staff from September of the second year. In year 3 we will hire another developer for a total of 4 developers, 3 customer support and success staff, and 1 part time marketing staff. This decision was made to support the growing number of labs each year and so we can carry out the development tasks effectively without delays, while our customer support team is not short staffed.

### Marketing

We are going to spend 10% of our total operating revenue on marketing our product each year.  
[24]

### Cloud Services

Labline will be entirely hosted on cloud infrastructure using AWS, the services we will use are EC2(application server hosting), RDS PostgreSQL(database management), S3 (file storage), Route53(DNS management), and CloudWatch(monitors and logging). These are essential services we will need to use to ensure the system can be scalable and secure.

Our projected costs for cloud services based on the expected usage are:

In year 1 we will spend €1,200 per month, in year 2 we will spend €1,500 per month and in year 3 we will spend €2,000. We have based these projections on the growth in traffic, lab adoption over the 3 years and monitoring needs. For the costs for year 1 we are assuming modest usage across the 10 labs, while for the costs of year 3 we are assuming higher usage since we will have 25 labs onboarded.

**Technical services and admin costs**

We will be spending €11.60 per user per month on Office365 which is essential for our business and staff. We will also be spending €13.53 per user per month on jira, which is another essential tool we need for more efficient project management. GDPR consultancy for compliance will cost €140 per month for our business. **[25] [26] [27]**



## References

[1] Ovcharenko, D. (2024, January 9). *"Perfect Software Developer Team: Roles and Responsibilities"*. Alcor. <https://alcor-bpo.com/10-key-roles-in-a-software-development-team-who-is-responsible-for-what/>

### Proposed Market

[2] *"How many universities are there worldwide?" 2025 Statistics*. <https://en.uhomes.com/blog/how-many-universities-are-there-worldwide#:~:text=According%20to%20statistics%2C%20there%20are%20about%2050%2C000%20universities%20in%20the%20world.https://www.unesco.org/en/higher-education>

[3] *"Higher education figures at a glance"*. UNESCO. [https://uis.unesco.org/sites/default/files/documents/f\\_unesco1015\\_brochure\\_web\\_en.pdfhttps://ultimatecollegelist.com/all/colleges/ireland](https://uis.unesco.org/sites/default/files/documents/f_unesco1015_brochure_web_en.pdfhttps://ultimatecollegelist.com/all/colleges/ireland)

[4] (2022, June 3). *"List of Colleges in Ireland"*. UltimateCollegeList. <https://ultimatecollegelist.com/all/colleges/ireland>

[5] O'Brien, C. (2023, September 25). *"Student-Staff ratios in universities "much-worse" than recession-hit 1980s, warns UCD president"*. IrishTimes <https://www.irishtimes.com/ireland/education/2023/09/25/student-staff-ratios-in-universities-are-much-worse-than-recession-hit-1980s-warns-ucd-president/>

[6] <https://d1eipm3vz40hy0.cloudfront.net/pdf/cxtrends/cx-trends-2020-full-report.pdf>

### Supporting Industry Trends

[7] *"Edtech Market Size, Trends Analysis, Forecast 2034"*, Marketresearchfuture. <https://www.marketresearchfuture.com/reports/edtech-market->

16213#:~:text=Edtech%20Market%20Size%20was%20valued,period%20(2023%20%2D%202032).

[8] “Key Facts and Figures 2023/2024”. HEA. <https://hea.ie/statistics/data-for-download-and-visualisations/key-facts-figures/>

[9] (2022, August 18th). “Computer Science now the fastest growing degree subject, new data reveals”. BCS. <https://www.bcs.org/articles-opinion-and-research/computer-science-now-the-fastest-growing-degree-subject-new-data-reveals/#:~:text=Computer%20Science%20now%20the%20fastest%20growing%20degree%20subject%2C%20new%20data%20reveals,-18%20August%202022&text=Record%20numbers%20will%20start%20computer,subject%20group%2C%20new%20figures%20show.>

[10] “Learning Management Systems: Current Trends and Challenges in Higher Education”. EducationTechnologyInsights. [Learning Management Systems: Current Trends and Challenges in...](#)

[11] Morgan, G. (2024, March 14). “Observations on the Higher Education LMS Market”. OnEdTech. <https://onedtech.philhillaa.com/p/observations-on-higher-ed-lms-market>

[12] “Queue Management System Market Growth Opportunities and Forecast by 2031”. TheInsightPartners. <https://www.theinsightpartners.com/reports/queue-management-system-market>

<https://www.globenewswire.com/news-release/2025/02/06/3022159/0/en/Queue-Management-System-Market-to-Reach-USD-1-20-Billion-by-2032-Research-by-SNS-Insider.html#:~:text=%E2%80%9CThe%20Queue%20Management%20System%20Market,%25%20from%202024%20to%202032.%E2%80%9D>

### Pricing Model

[14] Microsoft Word - SALARY SCALES FOR THE WEB 01.10.2020 (2).docx

### Competitors

[15] <https://www.zendesk.com/service/ticketing-system/>

[16] <https://qless.com/>

[17] <https://support.zendesk.com/hc/en-us/articles/4408887031322-How-does-Zendesk-send-email-notifications>

### Primary and Secondary Research

[18] <https://www.smartsurvey.co.uk/blog/usability-testing-how-surveys-can-improve-it>

[19] [https://www.researchgate.net/publication/344699426\\_Measuring\\_Impact\\_Student\\_and\\_Instructor\\_Experience\\_Using\\_an\\_Online\\_Queue](https://www.researchgate.net/publication/344699426_Measuring_Impact_Student_and_Instructor_Experience_Using_an_Online_Queue)

<https://docs.google.com/forms/d/16fEUUH5vGgRBaBeQp2NQzWf6uEnbDMFwF9o4ILL2WIE/edit>

[21] <https://experts.illinois.edu/en/publications/adoption-of-an-online-queue-system-for-education-a-case-study>

[22] <https://dl.acm.org/doi/pdf/10.1145/3197091.3205837>

[23] AIB loan calculator - <https://aib.ie/business/loans-and-finance/quick-quote-calculator>

### Marketing expenses

Spend 12-20% of annual revenue on marketing expenses.

Costs [24] <https://www.startuploans.co.uk/support-and-guidance/business-guidance/marketing/how-much-money-should-you-spend-on-digital-marketing-for-your-start-up>

### Technical services:

[25] <https://www.microsoft.com/en-ie/microsoft-365/business#layout-container-uid4d2d>

LabLine - Final Delivery

Office365 = 11.60 per person per month annual subscription

**[26]** Jira - <https://www.atlassian.com/software/jira/pricing>

13.53 per user per month

GDPR Consultancy

**[27]** [https://gdprhero.com/prices/?utm\\_source=chatgpt.com](https://gdprhero.com/prices/?utm_source=chatgpt.com)