D798 - Task 1 - Attempt 2

Silver Alcid - #011933973

8/22/2025

A1. Identify a precursor technology to this system from at least 10 years ago.

A precursor technology to the Virtual Learning Environment (VLE) from more than ten years ago is the Learning Management System (LMS), with platforms like Blackboard and Moodle that were commonly used in the early 2000s. These systems were an early step into online education, but they were fairly limited compared to what is available today.

The main purpose of an LMS was to provide basic course delivery. Instructors could upload lecture notes, post assignments, and set up quizzes for students to complete. Most of these systems ran on local servers owned by the school, which meant institutions had to dedicate IT staff to handle maintenance and troubleshooting (Almarashdeh and Alsmadi 2025).

Interaction within these early systems was also very basic. Communication usually came through text-based discussion boards or email, with no built-in video conferencing, real-time collaboration tools, or flexible cloud support (Faig 2025). Because of these limitations, LMS platforms can be seen as the direct precursor to modern VLEs, which expand on those foundations by adding interactivity, scalability, and integrated cloud-based features.

A2. Describe how the system you have chosen has improved upon the previous technology.

The Virtual Learning Environment (VLE) has made big improvements over the older Learning Management Systems (LMS). One of the most important changes is that VLEs run on cloud infrastructure, which makes them more scalable, accessible, and reliable compared to the server-based systems schools used in the past. Institutions no longer have to maintain expensive on-site servers or depend heavily on IT staff, since the cloud handles updates, backups, and performance automatically.

Another improvement is in communication and learning tools. While LMS platforms mainly offered text discussion boards and email, VLEs now support real-time video conferencing, live chat, collaborative workspaces, and mobile access (Faig 2025). They also include analytics that let instructors monitor participation and progress more closely. These changes make the learning experience more interactive, flexible, and personalized for both students and teachers.

A3. Compare the cost of implementing the old system to that of the new system.

The older LMS required schools to purchase physical servers, costly software licenses, and maintain full-time IT staff, leading to annualized costs of $250,000–$400,000+ for mid-sized institutions. These expenses were not only high upfront but also grew with ongoing maintenance, hardware replacements, and troubleshooting, making the systems financially challenging, especially for smaller schools.

Modern VLEs in 2025 rely on cloud-based infrastructure, eliminating the need for expensive on-site hardware and reducing staffing requirements. Institutions instead pay subscription and hosting fees that scale with the number of users, averaging $200,000–$350,000 annually for mid-sized universities. While costs remain significant for larger implementations, VLEs are generally 20–40% more affordable than traditional LMS, offering predictable budgeting, flexible scalability, and reduced IT burden.

Learning Management Systems (LMS, early 2000s – on-premise model)

Hardware (servers, storage, networking):

• Mid-sized institution required 3–5 dedicated servers.

• Approx. cost per server (enterprise-grade): $10,000–$20,000 each.

• Total hardware upfront: $50,000–$100,000.

Software licensing:

• Blackboard licenses in early 2000s ranged $50,000–$200,000 annually, depending on user count.

• Moodle was free/open-source but required paid customization and support ($20,000–$50,000 annually).

IT staffing:

• At least 1–2 full-time IT administrators.

• 2025-adjusted salaries: $80,000–$100,000/year per staff member.

Maintenance & upgrades:

• Hardware replacement cycle every 3–5 years: $30,000–$50,000 per cycle.

• Downtime risks increased operational costs.

Estimated annualized cost (mid-sized university): $250,000–$400,000+.

Virtual Learning Environments (VLE, 2025 – cloud-based model)

Subscription/licensing (SaaS model):

• Most providers (such as Canvas, Brightspace, Blackboard Ultra) use per-user pricing.

• Typical range: $5–$20 per user per year.

• Mid-sized institution (10,000 users): $100,000–$200,000 annually.

Cloud hosting (scalability & storage):

• Average cost of cloud services (compute + storage + CDN): $0.50–$2 per user/month.

• For 10,000 users: $60,000–$240,000 annually.

IT staffing:

• Reduced compared to LMS. Typically 0.5–1 full-time staff for admin/training.

• 2025 salary estimate: $50,000–$70,000 annually.

Additional integrations & support:

• Analytics, proctoring, and third-party app integrations: $20,000–$50,000 annually.

Estimated annualized cost (mid-sized university): $200,000–$350,000.

A4. Explain whether the new system is financially beneficial to its users. Describe what basis you used to determine your answer.

The VLE is financially beneficial because it shifts away from the heavy upfront costs that came with the LMS. Schools using an LMS had to buy and maintain their own servers, pay for software licenses, and keep IT staff on hand to manage the system. With the VLE, most of those expenses are reduced since it runs on the cloud. Institutions usually just pay subscription fees that scale with the number of users, which makes it easier to manage budgets and avoid big infrastructure investments (Faig 2025). For students, the VLE can also save money by giving access to digital materials and remote learning options, which lowers costs tied to textbooks, commuting, and even housing in some cases.

My conclusion is based on comparing the overall cost structures of LMS and VLE systems. The LMS model demanded larger upfront spending and ongoing maintenance, while the VLE spreads out costs in a more predictable way through cloud services. Because of this, the

VLE ends up being more financially sustainable for schools and creates savings and convenience for students too.

B1. Identify the type of hardware required for the system, including whether it is proprietary.

The hardware needed to run a VLE is fairly standard and does not usually require anything proprietary. On the institution side, the main requirements are servers to host the web application, databases, and storage. In most cases, these are provided by cloud service providers like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud, so the school does not need to buy or maintain physical servers on campus. For redundancy and performance, the servers are typically virtualized and spread across multiple data centers.

On the user side, students and instructors only need basic hardware like laptops, tablets, or even smartphones with an internet connection to access the VLE. Peripherals such as webcams, microphones, and headsets are also useful for video conferencing and live interaction (Almarashdeh and Alsmadi 2025). None of this hardware is proprietary, which makes the system more accessible because it works with common, widely available devices.

B2. Identify the type of data center or cloud infrastructure required for scalability and performance.

For scalability and performance, the VLE would rely on a public cloud infrastructure. Services like AWS, Microsoft Azure, or Google Cloud are ideal because they provide elastic computing resources that can automatically scale up or down depending on the number of users. This flexibility is important since the system might only have a few classes running at one time but could also need to handle thousands of users during peak periods like exams. Using cloud infrastructure also improves performance and reliability because these providers distribute workloads across multiple data centers. That means if one server or location goes down, the system can still stay online through redundancy. The cloud also offers built-in tools for load balancing, content delivery networks (CDNs), and database replication, which all help the VLE deliver fast response times and a smoother experience for students and instructors.

B3. Identify the operating system that will be used for the system and explain the rationale for this choice.

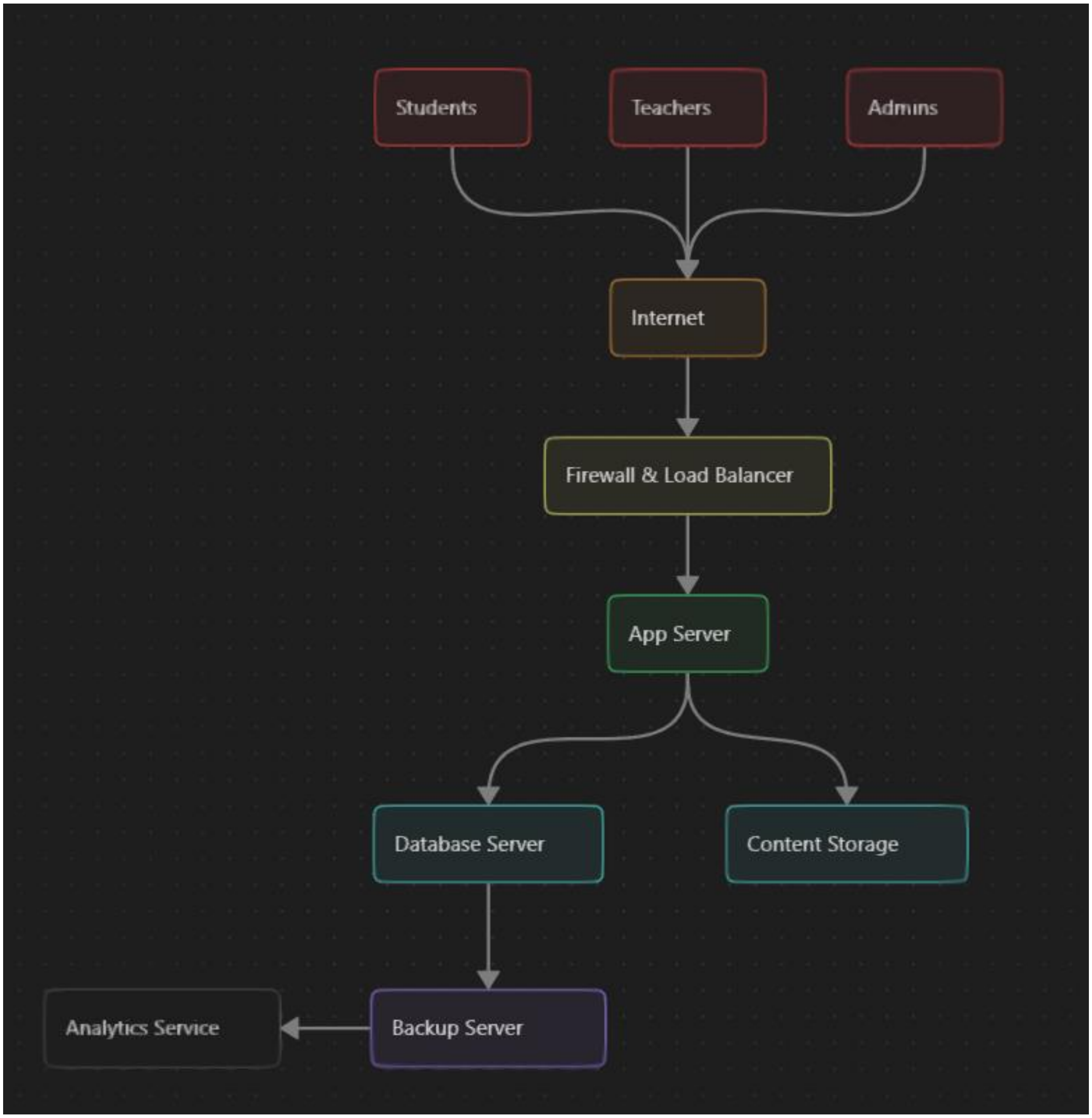
For the server side of the VLE, the best choice of operating system would be Linux, specifically something like Ubuntu LTS. This distribution is very popular in cloud environments because it is stable, secure, and lightweight compared to some alternatives. It also comes with long-term support, meaning updates and patches are delivered consistently over several years, which reduces the risk of downtime or system failures.

Linux also works well with the types of tools that a VLE depends on. It supports common web technologies such as Apache, Nginx, Node.js, and Python, as well as major databases like MySQL and PostgreSQL. These integrations make it easier for developers to maintain and scale the platform. Another big benefit is that Linux does not require licensing fees, which helps keep costs low compared to server systems like Windows Server.

For students and instructors, the operating system is not really a barrier because the VLE is accessed through a web browser. This means it can run on almost any device, whether it’s Windows, macOS, Android, iOS, or even ChromeOS. Users just need a stable internet connection and a modern browser, which makes the system more flexible and widely accessible.

Overall, using Linux on the server side keeps the VLE affordable, secure, and efficient, while the browser-based design ensures that anyone can access it regardless of what device they have. This combination of stability for the backend and flexibility for the users makes Linux the most practical choice for supporting the system.

B4. Provide a visual diagram of the system architecture using a design environment of your choice (e.g., Visio, Figma, Canva). Explain how the different components will communicate and interact with each other.



In this design, the VLE begins with the users (students, teachers, and administrators) who connect to the system through the internet. Their requests first pass through a firewall, which protects against unauthorized access, and then through a load balancer. The load balancer's role is to distribute incoming traffic evenly across available servers so that no single machine becomes overloaded and the system remains responsive.

Once traffic passes this security and distribution layer, it reaches the web and application server. This is the core of the VLE where most of the processing happens. The web/app server handles authentication, delivers course content, and manages user interactions like quizzes, assignments, and communication tools. To support these functions, the application server communicates directly with the database server. The database holds user information, enrollment data, grades, and other structured records.

For larger files, such as videos, lecture slides, and other media resources, the web/app server also connects to a separate content storage system. This ensures that heavy files can be delivered efficiently, often through cloud-based object storage and content delivery networks (CDNs). To maintain reliability, the database server regularly replicates its data to a backup server. This way, if the primary database fails, the system can restore information quickly without major data loss.

Finally, the system includes an analytics service, which gathers data from both the application server and backups. This service provides insights into system performance and user activity, such as tracking participation, monitoring engagement, and generating reports for administrators. Together, these components form a reliable, scalable, and secure system where each part communicates through secure connections to keep the VLE functional and accessible at all times.

B5. Describe the impact on the overall system if one component fails.

If one component of the VLE fails, the impact will depend on which part of the system is affected. For example, if the web/app server goes down, users would not be able to log in or access any courses since this is the main entry point for all requests. Similarly, if the database server fails without a backup, the system would lose access to important records like student accounts, grades, and enrollment data, which would make the VLE unusable until it is restored.

A failure in content storage would limit access to large files such as lecture videos or assignment uploads. The VLE might still load, but the user experience would be incomplete because key resources would be missing. If the load balancer fails, traffic could not be distributed properly, which could overload a single server and lead to downtime. Even the analytics service, while not critical for basic functionality, would still cause problems if it failed since administrators would lose visibility into performance and engagement data.

Overall, every component plays a role in keeping the system reliable. A single failure can disrupt services for users, but the impact can be minimized with backups, redundancy, and failover systems in place.

B6. Describe two strategies that will be implemented to ensure performance and reliability.

One strategy to ensure performance and reliability is to use **load balancing with redundancy**. By having multiple application servers running at the same time, the load balancer can spread out traffic so no single server gets overwhelmed. If one server fails, requests are simply redirected to the others, which keeps the system available for users. This setup not only improves performance during peak usage but also makes the system more fault-tolerant.

Another strategy is to use **regular backups and database replication**. The database is one of the most critical parts of the VLE, and if it goes down or data gets corrupted, the system would not function properly. Replicating the database across multiple locations and running automatic backups reduces this risk. If the primary database fails, a replica can take over quickly, which minimizes downtime and protects student data.

B7. Analyze the system's behavior under loads of 10 users, 1000 users, and 1 million users. Propose two methods that could be used to scale the system effectively.

10 concurrent users

- One small app instance and a single managed database are sufficient.
- Latency stays low; CPU, memory, and DB IOPS remain well under capacity.
- No special tuning beyond HTTPS, basic caching, and nightly backups.

1,000 concurrent users

- A load balancer with 3–6 app instances handles traffic comfortably.
- Hot paths (login, course feed, grade lookups) start to stress the database; add a connection pool and a small in-memory cache (e.g., Redis) for session/state and frequently read objects.
- Store and serve media from object storage with a CDN to avoid saturating app nodes.
- Expect acceptable latency if DB read/write contention is controlled and static assets are offloaded.

1,000,000 concurrent users

- Requires distributed, multi-region design. App tier must be stateless and horizontally scaled to hundreds/thousands of pods/VMs behind global load balancing
- Aggressive edge caching (CDN) for static and semi-dynamic content; signed URLs for media.
- Database becomes the primary bottleneck: use read replicas, sharding/partitioning, and asynchronous queues for non-critical writes.

- Background jobs (grading, analytics ingestion, video processing) move to a queue/worker model to protect the request path.
- Apply rate limiting and backpressure to preserve core functionality under surges.

One method to scale the VLE effectively is by making the application stateless and using horizontal scaling with autoscaling features. By running the application in containers, such as on Kubernetes, new instances of the web/app server can be added or removed automatically depending on demand. This way, the system can handle everything from a few dozen users to thousands or more without performance issues. Since the app is stateless, any instance can serve any user session, and the actual session data or user state is stored in external services like a database or Redis. Combined with a load balancer and replication of the database across availability zones, this approach keeps the system both scalable and fault-tolerant.

Another method is to use caching at multiple levels. At the edge, a content delivery network (CDN) can serve static files such as videos, images, and course documents, reducing the load on the main servers.

At the application level, tools like Redis can cache user sessions, course data, and frequently accessed queries for short periods. This dramatically cuts down on database queries and speeds up response times. By serving a large portion of requests from caches rather than the database or core application servers, the VLE can support much higher user loads while still delivering a smooth experience.

B8. Identify one potential security risk of the system. Provide one strategy to prevent a cyberattack and one strategy to respond to a cyberattack.

B9. Identify one potential energy efficiency deficit in the system and one method to improve it.

One potential security risk for the VLE is unauthorized access through stolen or weak user credentials. Since students, teachers, and administrators all log in through the same system, a single compromised account could expose sensitive information such as grades, personal details, or course materials. If an administrator account were breached, the attacker could even gain access to the entire platform's settings and data.

A strategy to prevent this kind of attack is to require strong authentication methods. This includes enforcing complex passwords, adding multi-factor authentication (MFA), and monitoring login attempts for unusual behavior. By making it harder to log in with just a stolen password, the risk of account compromise is reduced significantly.

If an attack does occur, the response strategy should focus on fast detection and containment. Suspicious activity can be flagged by intrusion detection systems and immediately lock down affected accounts. At the same time, system logs should be reviewed to trace what data may have been accessed, and users should be notified if their accounts were impacted. Once the threat is contained, patches and policy updates can be applied to prevent the same type of attack from happening again.

B10. Identify one potential sustainability deficit in the system and one method to improve it.

One potential sustainability deficit of the VLE is its dependence on large-scale cloud infrastructure, which consumes significant amounts of electricity and contributes to carbon emissions. Data centers that host the servers, storage, and networking equipment use a lot of energy, not only to power the systems but also for cooling. This environmental cost can grow as the number of users increases.

A method to improve sustainability is to select cloud providers that prioritize renewable energy and green data centers. Many leading providers, like Google Cloud and Microsoft Azure, have made commitments to carbon neutrality and are increasingly relying on wind, solar, and hydroelectric power. By hosting the VLE on platforms that use clean energy and optimizing the system for efficiency (like auto-scaling to reduce idle server use), the overall environmental footprint of the system can be reduced.

B11. Describe your plan to validate the stability, scalability, and security of the system.

To validate stability, scalability, and security, the VLE would go through a structured testing and monitoring plan. For stability, stress testing and load testing would be used to simulate real-world scenarios and ensure the system can remain operational under different conditions. Continuous monitoring tools would also be in place to track uptime, error rates, and resource utilization so problems can be addressed before they cause outages.

For scalability, the system would be tested at increasing levels of load, starting from small groups of users and scaling up to thousands or more, to confirm that auto-scaling and load balancing are functioning as expected. These tests would also help identify performance bottlenecks, such as database queries or storage access, which can then be optimized. For security, regular vulnerability scans, penetration testing, and audits would be conducted. Multi-factor authentication and encryption would be validated to ensure user data is safe. Incident response drills would also be included so the team is prepared to react quickly if a breach occurs.

C. Write a reflection essay on the design process of your system that includes all of the following: the reasoning behind your choice of system design, other designs you considered, roadblocks you encountered and how you solved them, what you would change if you had the means to do so.

When I began working on the VLE design, my main goal was to create a system that felt both accessible and scalable. Online education has become essential, and I wanted the design to reflect that shift. Choosing a cloud-based approach made the most sense because it reduces the cost of infrastructure while still offering high reliability and performance. I leaned toward non-proprietary hardware and open systems because they're affordable and easier for institutions to adopt.

I did think about other approaches before settling on this design. One option was building the VLE entirely on local, on-premise servers. That idea seemed appealing at first because it would give schools more direct control. However, the costs of hardware, energy, and staff would make it unsustainable. Another option was to design something closer to a traditional LMS, focusing only on file uploads, quizzes, and grade tracking. The problem with that choice is that it doesn't fully capture the strengths of today's interactive learning systems. A modern VLE needs features like video conferencing, real-time collaboration, and analytics to stay relevant.

There were some roadblocks along the way, especially when I thought about scalability. At first, it was hard to picture how the system could handle a sudden jump from a few dozen users to thousands, or even millions, without breaking down. Researching cloud strategies like caching and load balancing helped me understand how to distribute traffic and reduce bottlenecks. Another challenge was figuring out how to make the system diagram clear.

My first draft had too many details crammed in, and it felt cluttered. I solved that by focusing only on the core components that best showed how the system works together. If I had the means to expand the design, I would add more advanced features. For example, an AI-driven learning assistant could personalize courses for individual students, or automated grading could save instructors time. I would also improve sustainability by monitoring energy use in real time and shifting workloads to greener data centers. These changes would make the VLE not just scalable and reliable, but also more intelligent and environmentally responsible. Even so, the design I created serves as a strong foundation that institutions could realistically implement today.

References

Faig, N. "Digital Learning in the 21st Century: Trends, Challenges, and the Evolution of Digital Learning Environments." Frontiers in Education, 28 Mar. 2025, www.frontiersin.org/journals/education/articles/10.3389/feduc.2025.1562391/full.

Almarashdeh, I., and F. Alsmadi. "Evolution of Learning Management Systems: An Insight into the Current Trends." International Journal of Emerging Technologies in Learning, vol. 17, no. 4, 2022, pp. 31-47. https://files.eric.ed.gov/fulltext/EJ1345322.pdf