VIA University College

Project Description

Semester Project 3

Group 3

Students

Guillermo Sanchez Martinez (355442)

Piotr Wiktor Junosz (355502)

Halil Ibrahim Aygun (355770)

Alexandru Savin (354790)

Eduard Fekete (355323)

Supervisors

Joseph Chukwudi Okika (JOOK)

Jakob Trigger Knop (JKNR)

Character Count: 13623

Word Count: 1839

Software Technology Engineering

3rd Semester

December 15, 2025

Contents

1 Problem Domain

1.1 Education problem

Education is one of the most important aspects of life in civilized societies (NAICU - NAICU - Improves Quality of Life, 2025). Humans need knowledge in order to function in today's world — from simple communication to career growth. Knowledge can improve quality of life in many ways (Burke, 2023). "Nowadays around 40% of the global population does not have access to proper education in a language they understand" (PTI, 2025).

Many countries and communities struggle with gender inequality in educational opportunities, which ultimately results in educational disparities and economic inequality. Men and women should be able to choose their educational paths without being constrained by gender norms or stereotypes (Gender Equality in a Changing World, 2025).

In addition to unequal access, gender inequality is also visible in the socialization and education of boys and girls, even when both have equal formal opportunities. Gender norms and stereotypes at early stages influence the types of skills children are motivated to learn and the careers they are likely to pursue. For example, international test scores show that around age ten, girls read more effectively than boys, while boys perform better in mathematics and science (OECD, 2022). These patterns reflect not only differences in ability but also the effects of socialization, classroom processes, and cultural expectations about "appropriate" fields of study for each sex.

These inequalities are also shaped by intersecting variables, including socioeconomic status, physical ability, and social status (OECD, 2020).

1.2 Current Situation

Digital free-for-all educational platforms have been on the rise in recent years because they can provide knowledge to anyone with an internet connection and a device, thus minimizing educational inequalities (Duolingo, 2025).

However, many currently operating platforms focus on different aspects of education, such as language learning, coding, mathematics, or science (Singh et al., 2015). This leaves gaps and does not provide a consistent learning experience for users. Additionally, some platforms offer little free content, which can be a significant barrier for low-income users (World Bank Group, 2025).

Despite rapid advances in digital technologies, a major challenge persists: 2.6 billion people remain offline (Poggi, 2025). The digital divide is a major barrier to economic growth and sustainable development, with only 27% of the population in low-income countries having access to the internet, compared to 93% in high-income countries (Poggi, 2025).

1.3 Stakeholders

The primary stakeholders in the field of education include both learners and knowledge providers worldwide. Learners can be individuals of all ages, backgrounds, and locations who seek to acquire new skills or knowledge. Knowledge providers can include educational institutions, teachers, tutors, and online platforms that offer educational content and resources. Other stakeholders may include governments, non-profit organizations, and businesses interested in promoting education and improving access to learning opportunities.

---

2 Problem Statement

2.1 Main problem:

How to provide easily accessible learning platform?

2.2 Sub-questions:

- Is there any way to speed up the learning process?
- How to make the learning process efficient and user friendly?
- How to ensure that all genders can acquire proper knowledge?
- What design principles can be used to make digital learning app more convenient?
- How can accessibility features (e.g., screen readers, voice commands, adaptive interfaces) be integrated for users with disabilities?
- How can trust in the system be established and maintained?
- How can the correctness of knowledge be ensured?
- How to provide and maintain security within the app?

---

3 Delimitation

The project focuses on helping individuals who cannot access formal education. This may be due to income reasons or geographic constraints. Men and women will have equal opportunities on the digital platform, since everyone should be able to pursue education as they wish. To ensure high-quality, accurate content, courses will be reviewed by experts and AI.

The project is limited to the development of a simple digital distributed learning platform for students, teachers, and its administrators. Features such as high-scale performance, complete accessibility, AI-based learning, and enterprise-level security are not part of it.

Delimitations Related to Sub-questions

Speeding up the learning process: The system will not be based on any researches evaluating optimal learning speeds. It will provide a basic platform where exercises can be accomplished and tracked.

Efficiency and friendliness of the learning process: Intuitive navigation and role-based design will be implemented. Deep UX testing, advanced personalization, or industry-level research of design is beyond the scope.

Gender neutrality in learning: The platform won't include gender-specific functionality. The platform will assume equal opportunity and neutrality for every user.

Principles of convenience design: The platform will use common UI design patterns suitable for desktop learning solutions. Research and comparison of other design philosophies (gamified vs. traditional layout) won't be cunducted.

Accessibility features (screen readers, voice commands, adaptive interfaces): The application will provide only limited accessibility. Advanced adaptive interfaces won't be implemented.

Active pursuit of learning goals: The system will offer basic monitoring of progress. High-level motivational features (gamification, AI reminders) will not be part of it.

Trust in the system Trust will be addressed primarily by role-based access control and secure authentication. Broader features such as institutional certification or pedagogical authentication are out of scope.

Correctness of knowledge: Correctness of the course material will be assumed. The system will not include AI fact-checking, peer-review workflows, or third-party verification of material.

Security Security will be guaranteed on a low level (password hashing, simple role-based permissions). Advanced protection (multi-factor authentication) is beyond the project.

---

4 Choice of Methods

4.1 Knowledge and Data Collection

To identify the educational and technical issues of digital learning platforms, we will:

- Review existing platforms (e.g., Coursera, Duolingo, Udemy) to ascertain their strengths and weaknesses.
- Review academic reports and papers on the digital divide and language learning strategies.
- Identify user needs from informal questionnaires and discussions within the project team.

4.2 Analysis and Modelling

For distributed system planning and design, UML diagrams will be used to model system functionality and interactions.

- Architectural patterns such as client–server and REST-based architectures will be part of the design.
- Threat modelling will be conducted and potential security threats in authentication and data communication will be analysed.

4.3 Design, Construction and Implementation

The system will be developed using standard software development methods:

- Agile methodology with short iterations to ensure continuous progress and responsiveness.
- UI/UX prototyping with Figma for wireframing and user flows prior to coding.
- Backend services implemented in Java/C# with RESTful web services.
- Authentication and authorization mechanisms to ensure secure access.

4.4 Testing

The testing will be carried out continuously during the project to ensure robustness and functionality:

- Unit testing (Java: JUnit / C#: NUnit).
- Integration testing to check interactions between client and server.
- Manual testing of UI components and learning flows for usability and accessibility checks.
- Security testing with emphasis on authentication and authorization.

4.5 Planning and Management

To facilitate organized collaboration and progress monitoring:

- Git with GitHub for version control, feature branching, and code reviews.
- Task distribution and workload management will be done using a Kanban board (Figma/GitHub Projects).
- Regular meetings to check progress, resolve problems, and plan next steps.
- Documentation will be written in a formal academic style, with correct referencing and adherence to plagiarism standards.

---

5 Time Schedule

5.1 Final Deadline

Date: December 19, 2025

6 Project Timeline

Date / Period	Milestone / Activity	Details
Every Sunday	Weekly Reporting & Task Assignment	Submit progress report + assign new tasks
Weekly	Weekly Meeting	Checkpoint via Discord or at school
End of November 2025	Completion of Formal Project Part	Finish writing & documentation for review
December 19, 2025	Final Deadline	Submission of full project

6.1 Milestones

1. **Weekly Reporting and Task Assignment**
   When: Every Sunday
   Details: Submit a weekly report on the project's progress and assign new tasks for the upcoming week to maintain continuous progress and team accountability.
2. **Weekly Meeting**
   When: Once per week
   Platform: Meetings will be conducted either via Discord or at school.
   Purpose: These meetings will act as checkpoints to discuss progress, address challenges, and adjust tasks as necessary.
3. **Completion of Formal Project Part**
   Target Date: End of November 2025
   Details: Aim to complete the formal writing and documentation aspect by this date, allowing time for final revisions before the deadline.

6.2 Expected Time Commitment Based on 10 ECTS

Each student is expected to contribute a total of 275 hours to meet the 10 ECTS workload requirement, with increased hours in October and November to reduce the workload in December.

6.2.1 Breakdown of Hours for 10 ECTS

- **October:**
  - Weekly commitment: 16 hours per student
  - Total for October: $4 \times 16 = 64$ hours per student
- **November:**
  - Weekly commitment: 16 hours per student
  - Total for November: $4 \times 16 = 64$ hours per student
- **December (up to December 20):**
  - Remaining: 147 hours
  - Weekly commitment: $147 / 3 \approx 49$ hours per student
  - Total for December: $3 \times 49 = 147$ hours per student

6.2.2 Total Hours Calculation for 10 ECTS

- Total Hours: 275 hours per student
- Calculation: Total hours $= 10 \times 27.5 = 275$ hours

7 Risk Assessment

Risk	Likelihood	Severity	Normalized	Preventive Actions
Data storage issues	3	3	2.0	Secure datasets early, create backups, use redundant storage
Local server failure	3	3	2.0	Regular backups, spare devices, monitor hardware health
Synchronization failures between distributed parts	2	3	1.5	Use messaging queues, retries, health checks, test synchronization

8 References

- ReferencesBurke, C. (2023, October 4). Why Is Education Important? The Power Of An Educated Society. Unity Environmental University; Unity Environmental University. https://unity.edu/articles/why-education-is-important/
- Duolingo. (2025). Company Strategy Overview | Duolingo, Inc. Duolingo, Inc. https://investors.duolingo.com/company-strategy-overview-0
- Gender gaps in educational attainment and outcomes remain: Gender Equality in a Changing World. (2025). OECD. https://www.oecd.org/en/publications/gender-equality-in-a-changing-world_e808086f-en/full-report/gender-gaps-in-educational-attainment-and-outcomes-remain_33ea8a2f.html
- Lin, M.-H., Chen, H.-C., & Liu, K.-S. (2017). A Study of the Effects of Digital Learning on Learning Motivation and Learning Outcome. EURASIA Journal of Mathematics, Science and Technology Education, 13(7), 3553–3564. https://doi.org/10.12973/eurasia.2017.00744a
- NAICU - NAICU - Improves Quality of Life. (2025). Naicu.edu. https://www.naicu.edu/research/shaping-lives-and-anchoring-communities/improves-quality-of-life
- OECD. (2024). Early Learning and Child Well-being. OECD. https://www.oecd.org/en/publications/early-learning-and-child-well-being_3990407f-en.html
- Organisation For Economic Co-Operation And Development (OECD). (2020). PISA 2018 Results (Volume IV) : Are Students Smart about Money. Oecd.
- Poggi, A. (2025, April 4). The Digital Divide: A Barrier to Social, Economic and Political Equity | ISPI. ISPI. https://www.ispionline.it/en/publication/the-digital-divide-a-barrier-to-social-economic-and-political-equity-204564
- PTI. (2025, March 2). 40% global population doesn't have access to education in language they understand: UNESCO. Deccan Herald. https://www.deccanherald.com/world/40-global-population-doesnt-have-access-to-education-in-language-they-understand-unesco-3428194
- Singh, J., Fernando, Z. T., & Chawla, S. (2015). LearnWeb-OER: Improving Accessibility of Open Educational Resources. ArXiv.org. https://arxiv.org/abs/1509.02739
- World Bank Group. (2025, January 29). Digital Pathways for Education: Enabling Greater Impact for All. World Bank; World Bank Group. https://www.worldbank.org/en/topic/edutech/publication/digital-pathways-education-enabling-learning-impact

VIA University College

Project Title - Process Report

Semester Project 3

Group 3

Students

Guillermo Sanchez Martinez (355442)

Piotr Wiktor Junosz (355502)

Halil Ibrahim Aygun (355770)

Alexandru Savin (354790)

Eduard Fekete (355323)

Supervisors

Joseph Chukwudi Okika (JOOK)

Jakob Trigger Knop (JKNR)

Character Count: 9518

Word Count: 1474

Software Technology Engineering

3rd Semester

December 15, 2025

Contents

1 Introduction

The project of a Learning Platform was initiated with the goal of creating an accessible and user-friendly online environment for learners. This project focused on developing a distributed system in a team environment, leveraging various distinct technologies and methodologies to ensure effective collaboration and successful delivery.

---

2 Group Work

Many of our group processes stemmed as a continuation from out previous project experience. Some of the core values that drove our collaboration included open communication, mutual respect, fairness and democratic decision-making.

We based the group contract (TODO: REFERENCE) for the project on our previous contract with slight changes, for example to the point system. However, we did not have to make any major changes as our previous contract had already established a strong foundation for teamwork. This group contract outlined our core values, purpose, expectations, conflict resolution strategies, and accountability measures in the team. In the true agile and democratic spirit, we ensured to have mechanisms in place to be able to revisit the contract and modify it appropriately as the project evolved.

An important feature of the project's process was the role of a Product Owner (PO). The PO was responsible for managing the user stories, prioritizing the backlog, and ensuring that the team was aligned with the project goals. This role was crucial in maintaining a clear vision and direction for the project. Contrary to our previous experience, this role was set to be more settled and less rotating among team members. This was done to ensure consistency and a clear point of contact for the team regarding project requirements and priorities. The role of the PO was assigned democratically at the start of the project, and the team (TODO: COMPLETE LATER)

---

3 Project Initiation

The initial ideas we worked with included a Health Assistant, Bank System, and the Learning Platform. These were chosen as relevant candidates from a brainstorming session based on their potential impact and feasibility. After discussing the vision and scope of each idea, we held a democratic vote to select the final project. The Learning Platform was chosen due to its clarity of purpose and alignment with our skills and interests.

The alignment on the actual vision was more complicated than the initial idea selection. We had to ensure that everyone in the team agreed on the project's goals, target audience, and key features. This involved several discussions and compromises to reach a consensus that satisfied everyone.

One of the problems we faced during the initiation phase was deciding on the actual scope of the project. We were all juggling between thinking about it as an actual ambitious product versus a simpler prototype suitable for the course requirements. This problem was resolved by agreeing on scalable and flexible solutions that allow for both simple approaches and potential complex upgrades.

Another challenge was approaching the problem in a data-driven manner. Considering our mostly European backgrounds, we had to ensure that the project's aims were relevant and applicable on a global scale by researching, and not relying solely on our personal experiences and assumptions. This also included exploring various perspectives and shifting some of the focus on minorities and underrepresented groups in education. This aligned with our mission to improve the education, rather than just having business goals focusing on making profit off of profitable majorities.

---

4 Project Execution

The way of working in the project was a radical shift from our previous experiences. We adopted various aspects of Agile methodologies that fit our desired workflow. This included sprints, regular meetings, pair programming, Kanban but also our own adaptations to these practices.

Our main workflow was initially communicated and written down in the (TODO: REFERENCE). This document served to outline our processes and roles but also aligned us on semantics and definitions of key concepts. Even though this document served as our aim and theoretical perfection we could aim for, perfect adherence was not always possible and advantageous.

One of crucial aspects of our work was embracing asynchronicity, which might contradict with our vision of pair programming but was seen collectively as the means to achieving better productivity and work-life balance. Asynchronicity was also implemented on a level of subgroups - fx. different feature groups would collaborate within the group in real time while asynchronously reviewing and agreeing with the rest of the team on other issues.

4.1 Before Project Period

Because of daily school and work commitments, we mostly worked asynchronously in this period. Our sprints were one week long, starting and ending on Wednesdays. Each sprint would start around lunchtime as we wanted to have time to end the previous sprint collaboratively. Each sprint would start with a planning meeting where we would discuss the goals and tasks for the sprint. At the end of each sprint, we would review all features and tasks, and discuss any blockers or challenges faced during the sprint.

4.2 Project Period

Our work during the project period looked as follows:

4.2.1 Week perspective

We met every working day, weekends voluntary individually.

4.2.2 Day perspective

We preferred to work remotely as this reduced commuting time, allowed for more flexible working hours, but also prepared us for future remote work.

Each day started at 8:20 with a daily standup meeting where we discussed for each: - What was done - What will be done - Any blockers

After the standup, we would plan the day's work, which we considered one sprint in this dense period. Based on our framework, the purpose and focus of a sprint is to merely label and agree on which aspects of the Kanban board we focus on during the sprint. Contrary to Scrum, not all focus was put on putting tasks from TODO to DONE, but rather on deciding how far to push each task (allowing reviews to be done in a different sprint and similar).

Depending on the approach of each sprint, we would either be in a group call or split into channels for pair programming. With more iterations, we realized that collective calls were more effective as they allowed for quicker communication and synchronization on key issues; often problems would reappear for multiple people and having everyone listen to the solution saved time in the future.

Each day the calls lasted until around 16:30 with breaks for lunch and short breaks in between. Depending on everyone's situation, some days the work would continue later into the evening/night for those who preferred that.

4.2.3 Feature perspective

Working on a feature was usually for 1-3 people subgroups. The feature was started by discussing the definition of done, breaking down the tasks, and creating a branch for it. Each feature was practically a vertical slice of the product and the work was also often split into vertical slices (in order to reduce interpersonal dependencies). The aim of each feature team was to deliver a working feature compatible with main (which was evolving in parallel) by the end of the feature work. Each feature would have to be reviewed by someone least biased before merging to main.

Working with the team usually involved calling, sketching ideas, and drawing UML diagrams, while researching domain context and processing data from users. On a practical level, we used Visual Studio Code with Live Share for pair programming, GitHub for version control, Figma/FigJam for brainstorming and unrestricted diagramming, PlantUML inside VSCode for quick UML sketches, and Discord for screen sharing and communication.

---

5 Personal Reflections

5.1 Guillermo Sánchez Martínez

(TODO: Write a reflection for yourself)

5.2 Piotr Junosz

(TODO: Write a reflection for yourself)

5.3 Alexandru Savin

(TODO: Write a reflection for yourself)

5.4 Halil Ibrahim Aygun

(TODO: Write a reflection for yourself)

5.5 Eduard Fekete

(TODO: Write a reflection for yourself)

---

6 Reflect on Supervision

In the project team, we were often aligned on ideas and approaches, which made critical analysis and reflections on our work more challenging. The role of supervision thus became a crucial element of our project process. The supervisors often challenged our assumptions and decisions, ensuring that the project was kept improving based on proper processes and data, rather than just group consensus and momentum.

Each of our supervisor meetings had a clear agenda - acquiring external feedback and clarifying doubts. We did not focus on mere presenting of our progress but rather on seeking constructive criticism and guidance. This approach allowed us to identify potential pitfalls and areas for improvement that we might have overlooked without the supervisors.

Another aspect of the supervision was filling the technical gaps in our knowledge and project requirements. The supervisors always helped with long-term planning which could not have been done by the team alone due to the limited knowledge towards the beginning of the project.

(TODO: COMPLETE LATER, maybe something about frequency of meetings, communication outside meetings, etc.)

---

7 Conclusion

---

8 References

VIA University College

Project Title - Project Report

Semester Project 3

Group 3

Students

Guillermo Sanchez Martinez (355442)

Piotr Wiktor Junosz (355502)

Halil Ibrahim Aygun (355770)

Alexandru Savin (354790)

Eduard Fekete (355323)

Supervisors

Joseph Chukwudi Okika (JOOK)

Jakob Trigger Knop (JKNR)

Character Count: 12580

Word Count: 1878

Software Technology Engineering

3rd Semester

December 15, 2025

Contents

1 Abstract

This work focuses on the need for scalable and adaptable digital educational solutions by developing the Learnify system, a distributed software solution with the objective of ensuring seamless content delivery and user assessment. The primary goal is to make a durable multi-server solution that leverages the power of multiple programming languages for enhanced availability, data integrity, and optimized user response. This system was built with a polyglot microservices architecture, using programming languages such as Java and C# to ensure a strict interoperability implementation. Some of the key technical design choices are the use of gRPC for high-speed intra-service communication and HTTP for external client services, ensuring both speed and accessibility. Data storage is achieved using a PostgreSQL database, ensuring that security is taken into account using JWT authentication to overcome the potential dangers of distributed systems. This application development occurred using an iterative approach, centering around User Stories developed both using specialized analysis and interviews with the stakeholders. A functional prototype that can manage multiple user sessions concurrently in a distributed manner was developed. This prototype was verified to meet essential non-functional requirements, confirming the successful interoperability of components across Java and C#. Security compliance was established through the validation of salted password hashing algorithms using Argon2, ensuring robust data protection alongside reliable persistence in the PostgreSQL database. Moreover, the interface was validated for conformance with recognized accessibility guidelines to support complete usability by those with color vision defects. Finally, the project achieved a fully deployable distributed system, validating that the architecture provides a secure, stable, and operational foundation for scalability in the future of education.

---

2 Introduction

Acquiring new knowledge is essential part of human life and evolution. Living in the population equals communicating within it and that requires some minimum level of knowledge.(REFERENCE) The idea of mandatory education keeps its origin between late 1900s and early 2000s(REFERENCE), nonetheless around 40% of the global population still does not have access to proper education in a language they understand (PTI, 2025).

The aim of this project is on the ability to create a system which would be able to provide learning opportunities with main focus on simplifying the accessibility and exploring the idea of learning processes and its speed and efficiency. The goal is to also ensure security, knowledge correctness and deployment of the system.

With the pursuit of knowledge having been a cornerstone of human development for thousands of years, the incorporation of digital technologies into education is in perpetual evolution (Siemens, 2005). A widely accepted model for learning with digital technologies has not been identified, mainly because exponential increases in computing power and volumes of online information constantly redefine how users approach knowledge acquisition, processing, and retention (Haleem et al., 2022).

The approach of this project is to develop a distributed system implemented using at least 2 different programming languages, utilizing a database for data persistence, and adapting a hybrid communication strategy that includes technologies such as gRPC and HTTP.

---

3 Main Section

In the main section development experience of the “Learnify” e-learning platform is being documented. It describes the importance of essential phases that took place in the development process from the vision to a working software system. This chapter is organized around the core stages: Analysis, Design, Implementation, and Testing, ensuring that each were necessary while building the system.

This chapter starts by elaborating Analysis phase, which incorporates the functional and non-functional

requirements, use case diagrams, use case descriptions activity diagrams, threat model as well as the domain model of the system. The design section then focused on forming each parts of the system's architectures and database design, including the communication methods such as gRPC and HTTP. The Implementation stage converted the blueprints into components in two programming languages - C# and Java - that exhibit the integrated logic of the system. Finally, Testing were performed to ensure the system behaves according to the requirements and also meets the general objectives of the project.

3.1 Analysis

In the beginning of the technical documentation Analysis processes of the the distributed system - "Learnify" - are described. This stage applies the higher-level goals of the project to the real challenges that come with the multi-language distributed system.

For defining the scope of the proposed system with clarity, the initial idea was thoroughly refined using requirements. This involved the identification and definition of the functional requirements of the system as well as non-functional ones, including latency, scalability, and security, which are the most important requirements of the distributed learning system. In addition, several interviews with protentional clients were done in order to gather new data which helped creating and updating the requirements and then use cases. This stage of analysis also involved the use of structured modeling. The Use Case and Activity Diagrams were used for the representation of user interaction and system behavior, and the Domain Model defined the conceptual framework for the design of data persistence and service integration.

3.1.1 Requirements

3.1.1.1 Functional requirements

1. As a User, I want to register for an account so that I can access the platform.
2. As a User, I want to log in so that I can access the platform from my account
3. As a User, I want to view "My Courses" dashboard, so that I can see all the courses I am currently enrolled in.
4. As a User, I want to resume my active course from where I left off, so that I can continue learning from where i left.
5. As a User, I want to browse the "All Courses" catalog, so that I can discover new subjects to learn.
6. As a User, I want to filter courses by category or search query, so that I can find specific content quickly.
7. As a User, I want to unenroll from a course, so that I can remove content I am no longer interested in from my dashboard.
8. As a User, I want to view the Leaderboard, so that I can compare my progress with other learners.
9. As a User, I want to view my Profile, so that I can see my personal account details.
10. As a User, I want to complete Multiple Choice Questions in a learning step, so that I can test my knowledge.
11. As a User, I want to complete Fill-in-the-Blank exercises, so that I can practice recalling information.
12. As a Teacher, I want to create a new course draft, so that I can start building a new curriculum.
13. As a Teacher, I want to edit learning steps directly from the learning page, so that I can correct mistakes or improve content.
14. As an Admin, I want to view the Admin Panel, so that I can access administrative tools and settings.
15. As a Teacher, I want to edit course information, so that I can correct mistakes.
16. As an Admin, I want to view a list of waiting drafts, so that I can see which courses need approval.
17. As an Admin, I want to approve course drafts, so that they become available for students to enroll in.
18. As an Admin, I want to create new course categories, so that new types of courses can be introduced.
19. As an Admin, I want to add new languages, so that Teachers can create courses in other available languages.
20. As an Admin, I want to manage users' roles, so that I decide who is a teacher and admin.
21. As an Admin, I want to disapprove course drafts, so that they become unavailable for students to enroll in.

3.1.1.2 Non-functional requirements

1. The system must utilize at least two different programming languages
2. User passwords must be stored securely, using a strong, salted hashing algorithm before persistence
3. The system must use a robust database for all persistent data storage.
4. The system has to be deployable
5. The system has to be color blind people friendly

The requirements were taken from not only the initial vision of the system but also from interviews with real persons that could actually experience the system in the development stages and give relevant feedback to the team. The functional requirements were structured as "user stories" so that the team could maintain clear user perspective development, ensuring that every feature was directly tied to who is making what interaction with the system and how does the user is going to benefit from performing the action - why would he do it. This provided consistent framework for later analysis phases.

3.1.2 Use case diagram

3.1.3 Use cases and their related requirements

3.1.4 Use case description

3.1.5 Activity diagram

3.1.6 Threat model

3.1.7 Domain model

3.2 Design

3.2.1 System design

3.2.2 Architectural overview

3.2.3 Communication protocol design

*Interface Definition (IDL): Show snippets of your .proto files (if using gRPC).

API Specification: Briefly describe the HTTP endpoints (e.g., RESTful routes).

Protocol Choice: Explain why gRPC was used for internal communication vs. HTTP for external (or however you structured it).* ### Database design ### Class diagram design ### Communication Protocol Design: ### Data Persistence Design (maybe we should add some design related to this?):

ER Diagram: showing how data is structured in the database.

Consistency Model: Since it is distributed, mention how you handle data integrity across services.

3.3 Implementation

Focus on the "Polyglot" aspect (using 2+ languages) and the "Activities" involved.

3.3.1 Methods and tools

List the two languages (e.g., Go, Java, Python) and the database (e.g., PostgreSQL, MongoDB).

Justification: Explain why each language was chosen for its specific task. (e.g., "Go was selected for the backend service due to its concurrency handling...").

3.3.2 Server A Implementation (Language 1):

3.3.3 Server B Implementation (Language 2):

3.3.4 Server C Implementation:

3.3.5 Integration Logic:

Show how the two services "talk" to each other. Provide a code snippet showing the gRPC client/server handshake or the HTTP request handling.

3.4 Testing

3.4.1 Approach to testing

3.4.2 Tools and frameworks

3.4.3 What was tested

3.4.4 Method-level test case documentation

3.4.5 Benefits and bug detection

3.5 Result

The guidelines require you to support results with data, programs, or models.

3.6 Final Product Showcase: Screenshots of the "Learnify" app UI or console logs showing successful data processing.

3.7 Ethical Considerations:

Requirement: You must describe ethical considerations and how negative impacts are minimized.

Content: Discuss data privacy (GDPR), user consent, or the societal impact of the app.

---

4 Discussion

---

5 Conclusion and Recommendations

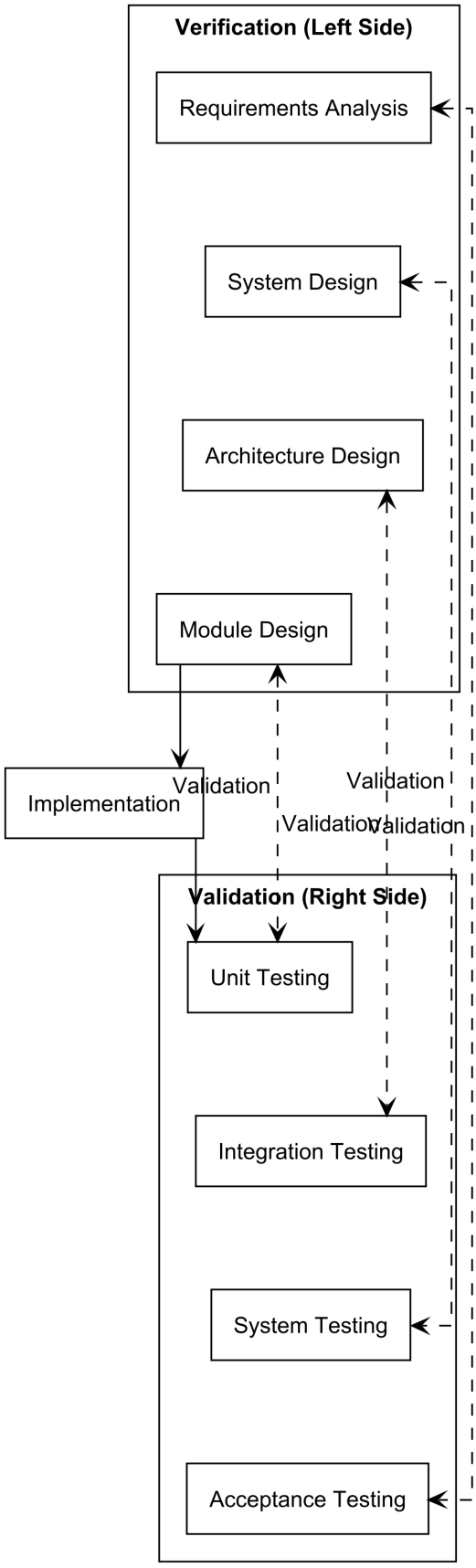
---

6 References

- project description sep3
- PTI. (2025, March 2). 40% global population doesn't have access to education in language they understand: UNESCO. Deccan Herald. https://www.deccanherald.com/world/40-global-population-doesnt-have-access-to-education-in-language-they-understand-unesco-3428194
- https://static1.squarespace.com/static/6820668911e3e5617c36c48c/t/682dadc9690ec5749004d96d/1747824073835/connec
- https://www.sciencedirect.com/science/article/pii/S2666412722000137

---

7 Test 123


Verification (Left Side)
Requirements Analysis
System Design
Architecture Design
Module Design
Implementation
Validation
Validation
Validation
Validation
Validation (Right Side)
Unit Testing
Integration Testing
System Testing
Acceptance Testing