Mathify

Technical Report II

Küng Pascal

Mettler Micha

Zehnder Jonas

Gonçalves Rafael

Thayananthan Ragavan

ZHAW-School of Engineering

IT22tb_WIN

Software Project 3

Lectured by Nina Schnatz and Kurt Bleisch

Abstract

Authors: Jonas Zehnder, Micha Mettler, Pascal Küng, Rafael Gonçalves & Ragavan Thayananthan

As global trends indicate a decline in mathematical proficiency among students, a team from the ZHAW School of Engineering developed Mathify, a novel educational platform aimed at revitalising mathematical learning for grades one through nine. **Methods:** The development of Mathify utilised Java to construct a robust, multi-tiered software architecture, integrating Javalin and Angular frameworks to ensure a seamless user experience. Strategic design and development were guided by comprehensive market analysis, detailed use case studies, and domain modelling, creating a platform that is both adaptable and pedagogically sound. **Results:** Mathify has significant potential in re-engaging students with mathematics, as it effectively merges gamified learning with the Lehrplan 21 syllabus, thereby improving both engagement and understanding. **Discussion**: The report concludes that Mathify is not only responsive to current educational challenges but is also scalable for future educational innovations. It sets a precedent for future digital learning platforms, suggesting expansive potential for broader pedagogical integration and ongoing adaptation to educational advancements.

Table of Contents

1 Initial Situation

In the rapidly evolving educational environment, it has become clear that challenges in mathematical learning among young students[1] are on the rise, leading to complex issues that span learners, educators, and parents alike [1]. The PISA 2022 assessment [2] [3] has highlighted a significant downturn in mathematics performance on a global scale, marking the steepest decline observed to date. This represents an average decrease of approximately 15 score points from 2018, which correlates to nearly a year's worth of learning, magnifying the crisis threefold compared to any past trends. Alarmingly, one in four 15-year-olds in OECD[2] countries now fall into the category of low achievers in mathematics. Despite students in Switzerland are performing slightly better [4] [5], the overarching message is consistent: foundational mathematical education is essential and cannot be overlooked.

In the wake of declining mathematics scores in PISA 2022, the demand for personalized mathematical education is spiking. Despite the vast number of online platforms available, there is room for these providers to more closely align with national syllabus[3] such as "Lehrplan 21" in Switzerland. Furthermore, incorporating gamification effectively and catering to various learning paces, along with providing supplemental materials for better comprehension, are areas ripe for development.[4]

1.1 Project Idea

Mathify[5] is designed as an educational platform that enhances mathematical skills for students in grades[6] one through nine through a gamified learning environment aligned with Switzerland's Lehrplan 21 syllabus. The platform introduces mathematical concepts progressively, catering to individual learning paces, and incorporates gameplay elements like levels[7], points, and badges[8] to motivate students. The platform aims to reduce mathematics anxiety by promoting a positive learning atmosphere where effort and learning from mistakes are valued over mere correctness. This approach encourages students to embrace challenges and enhances their understanding of mathematics. Mathify also supports educators and parents by providing tools that complement traditional teaching methods and offer real-time feedback on student

[1] See Glossary for a definition of this term.

[2] See Glossary for a definition of this term.

[3] See Glossary for a definition of this term.

[4] The validity of this statement is affirmed by a senior teacher assisting the team developing Mathify.

[5] See Glossary for a definition of this term.

[6] See Glossary for a definition of this term.

[7] See Glossary for a definition of this term.

[8] The badge system is not part of the prototype.

progress. This helps foster a deeper interest in and confidence towards mathematics among students.

1.2 Benefits of the Clients

Mathify integrates learning with interactive elements, providing significant benefits for users including students, parents, and educators. This section elaborates on how Mathify contributes to mathematical education through customisation, engagement, foundational development, flexibility, and educational partnership.

Customisation in Learning

Mathify functions as an adaptive learning tool, utilising advanced algorithms to match educational content with the learning pace[9] of each student. By aligning challenges[10] with the national syllabus and incorporating a mass-customised tutorial library[11], Mathify enhances user comprehension and retention. This adaptive approach ensures that the educational journey is not only aligned with but also surpasses the standards of digital mathematical education in Switzerland.[12]

Enhancing Engagement through Gamification

Mathify transforms the traditional learning process into an engaging and rewarding experience. The platform uses gamification techniques, such as badges, points, and interactive challenges, to make learning mathematics an enjoyable adventure. Each correct answer and completed milestone not only reinforces learning but also boosts the learner's enthusiasm for the subject, fostering a genuine appreciation for mathematics.

Foundational Mathematical Development

The early introduction of robust mathematical concepts is crucial for cognitive development in students. Mathify engages young learners from the first grade, providing

[9] Mathify dynamically adjusts its difficulty settings, scaling up the challenge with correct answer and easing the complexity when a student struggles, ensuring a personalised learning experience.

[10] The various modes are not part of the prototype.

[11] The tutorial library is not part of the prototype.

[12] cf. Chapter 1.3 “Competition Analysis”, which supports this statement.

a structured learning path that supports early mathematical understanding and sets the foundation for future academic and STEM[13]-related achievements.

Flexibility for Modern Lifestyles

Recognising the dynamic schedules of modern families, Mathify offers flexible learning solutions that integrate seamlessly into various daily routines. This flexibility ensures that students can engage in educational activities effectively, whether at home or on the move, without the constraints of fixed timetables.

A New Dimension in Educational Tools

For educators, Mathify serves as more than just a supplementary tool; it acts as a transformative educational partner. The platform enriches classroom instruction with digital elements and supports the customisation of exercises[14] to fit specific pedagogical needs. This capability allows teachers to craft a more personalised and effective learning environment, enhancing the educational experience for each student.[15]

1.3 Competition Analysis

In the realm of online learning platforms, each learning application presents distinct educational methodologies, creating a diverse landscape for product differentiation. The analysis that follows will examine the main three competitors in Switzerland, highlighting their strengths and weaknesses. This review is based on a variety of feedback sourced from the provider (cf. sources) and is further enriched by the author's direct experiences with these platforms.

1.3.1 ANTON

ANTON [6] is a digital learning platform offering a wide range of subjects for students of all ages. It stands out with its comprehensive content and interactive learning approaches, making learning engaging and fun. Particularly noteworthy are its diverse interactive exercises and games, aimed at motivating and supporting younger students. Teachers benefit from the ability to track student progress, assign tasks, and easily analyse results, simplifying lesson planning and feedback. Additionally, ANTON is easily

[13] See Glossary for a definition of this term.

[14] See Glossary for a definition of this term.

[15] The tailored features for teachers are not part of the prototype.

accessible on various devices, making it ideal for use both in the classroom and at home.

However, ANTON has areas for improvement. Since it is funded by a large EU project, the platform is somewhat rigid and cannot quickly adapt to rapid changes in educational requirements and technological advancements. This limitation hampers its ability to keep pace with dynamic educational demands. Another point of criticism is that the representation of the national syllabus of Switzerland on the platform is opaque and difficult for users to understand, posing challenges for teachers aiming to meet specifi c curriculum goals.

1.3.2 Khan Academy

Khan Academy [7] is a widely used online learning platform that covers a variety of subjects and topics, including mathematics, sciences, history, economics, and more. The platform offers an extensive collection of learning resources, such as interactive exercises, videos, articles, and practical applications, accessible to students of all ages and levels. One of Khan Academy's strengths is its clear alignment with national syllabus and standards. The platform provides a range of learning materials directly tailored to the content and requirements of the national syllabus of Switzerland, making it a valuable resource for educators implementing the curriculum. Additionally, Khan Academy offers personalized learning paths and adaptive learning, allowing students to choose their own learning pace and style. The platform tracks students' progress and adjusts the learning material accordingly to ensure they are challenged and able to reach their full potential.

Despite its strengths, Khan Academy also has some areas for improvement. Some users have reported that navigation on the platform is way too complex and very confusing, particularly for new users. Furthermore, additional interactive elements and gamification aspects is limiting student engagement and motivation.

1.3.3 Prodigy

Prodigy [8] is another popular mathematics learning application. It offers a comprehensive curriculum aligned with various educational standards, making it a versatile tool for both classroom and home use. The application stands out for its engaging gameplay and adaptive learning approach, where students solve mathematical problems to progress in the game world. This gamified aspect not only makes learning enjoyable but also motivates students to continue practicing and mastering mathematical concepts. Teachers appreciate Prodigy's ability to track student progress, identify areas of weakness, and provide targeted interventions. The application also offers resources for differentiated instruction, allowing teachers to tailor learning experiences to individual student needs. Furthermore, Prodigy provides real-time feedback to students, reinforcing correct answers and providing explanations for incorrect ones, fostering a deeper understanding of mathematical concepts.

Despite its many strengths, Prodigy has also some areas for improvement. As a result of its rapid growth and popularity, the application may experience occasional technical issues. Additionally, while Prodigy covers a wide range of mathematical topics, some educators may find that certain concepts or standards are not represented and often limit learners due to its predetermined content structure. Moreover, the content is not aligned with the national syllabus of Switzerland. Finally, Prodigy is the most expensive application of all of them with $74.95 for the cheapest package.

Conclusion

Despite all the strengths of its competitors, Mathify distinguishes itself in the online learning landscape with its quick adaptation to evolving educational requirements and technological advancements. It boosts user engagement through gamification and is fully aligned with the national syllabus Lehrplan 21, ensuring it meets Switzerland's educational standards. Mathify also offers intuitive navigation and technical stability, ensuring a seamless and reliable user experience, and this all at a competitive price. Therefore, Mathify is a superior choice for a comprehensive online learning solution.

1.4 Context Scenario

This chapter examines the practical applications of Mathify through the experiences of two distinct personas: Lea, an energetic second grader, and Matthias, an experienced mathematics teacher. This neutral observation outlines their interactions with the software, showcasing how it customises its features to meet diverse educational needs.

Persona and Goals

Lea is a second grader whose interest in traditional mathematics classes is waning. However, she is ambitious and receptive to a learning approach that incorporates elements of fun.

Matthias is a devoted mathematics teacher for grades four to six who struggles with the time-consuming process of creating personalised mock exams during the examination period. He is in search of an efficient solution that can streamline this process and enhance his students' preparation.

Actions and Interactions

When Lea uses Mathify, she starts by entering her name and educational level, initiating a tailored learning pathway. She chooses a game mode that specifies the type and difficulty of the mathematics challenges she will face. The software generates mathematical exercises based on her selections, with a scoring system that instantly reflects her progress, motivating her with each correct answer.

If Lea faces challenges, she has access to a tutorial library within Mathify that provides detailed explanations and examples to help her overcome difficult concepts, thus reinforcing her understanding and confidence.

Matthias finds significant relief in his hectic schedule through Mathify's capabilities. He quickly creates and distributes customised mock exams that meet the specific needs of his classes. The software automates the creation of these tests and adjusts the difficulty based on each student's performance, allowing Matthias to efficiently monitor and support his students' progress.

In essence, Mathify acts as a crucial educational tool for both learners and educators by automating and personalising the learning and assessment processes, thereby creating an effective and engaging educational environment in mathematics.

1.5 Profitability / Business Case Overview

The business case for Mathify presents a detailed strategy for the development and scaling of an educational platform aimed at enhancing mathematical skills among Swiss students. This strategy anticipates a transition from initial development stages to sustainable profitability within a five-year timeline, supported by various subscription models tailored for students, parents, and educational institutions.

Table 1 visualised below outlines the expected financial metrics over the first five years of operation, summarising revenue growth, investment in development, operational and marketing efforts, alongside overall profitability projections. These figures demonstrate Mathify's planned financial trajectory, reflecting strategic investments and anticipated returns. For a comprehensive breakdown of financial projections, competitive analysis, and detailed operational strategies, please refer to Chapter 7.4, entitled "Full Business Case", located in the appendix for further details.

Summary Table (all figures are in CHF)

Year	**Revenue: Basic**	**Revenue: Premium**	**Revenue: School**	**Total Revenue**	**Development Costs**	**Marketing & Sales Costs**	**Operational Costs**	**Profitability**
1	**0**	**0**	**0**	**0**	**20,000**	**50,000**	**10,000**	**- 80,000**
2	**1,000**	**50,000**	**12,000**	**63,000**	**20,000**	**100,000**	**10,000**	**- 67,000**
3	**2,000**	**250,000**	**60,000**	**312,000**	**20,000**	**250,000**	**50,000**	**- 8,000**
4	**3,000**	**500,000**	**180,000**	**683,000**	**50,000**	**300,000**	**100,000**	**+233,000**
5	**5,000**	**1,000,000**	**450,000**	**1,455,000**	**100,000**	**300,000**	**200,000**	**+855,000**

Table 1: Business Case Mathify Year 1-5

2 Analysis

This chapter provides a concise overview of the findings from the problem analysis for the Mathify application, detailing the use case diagram and the full spectrum of use cases, categorised as either fully dressed, casual, or brief. Additionally, it discusses the additional requirements that complement these use cases and concludes with a presentation of the domain model, capturing the essential concepts and relationships within the system. This structured approach offers a clear understanding of Mathify's framework and its capabilities to enhance educational experiences.

2.1 Use-Case-Model

Figure 1 below presents the use-case-model for the Mathify application, detailing the interactions between the system and its main users—students and teachers. This diagram systematically outlines essential user activities, including registration, mode selection, and exercise execution, illustrating a clear pathway through the application's features.

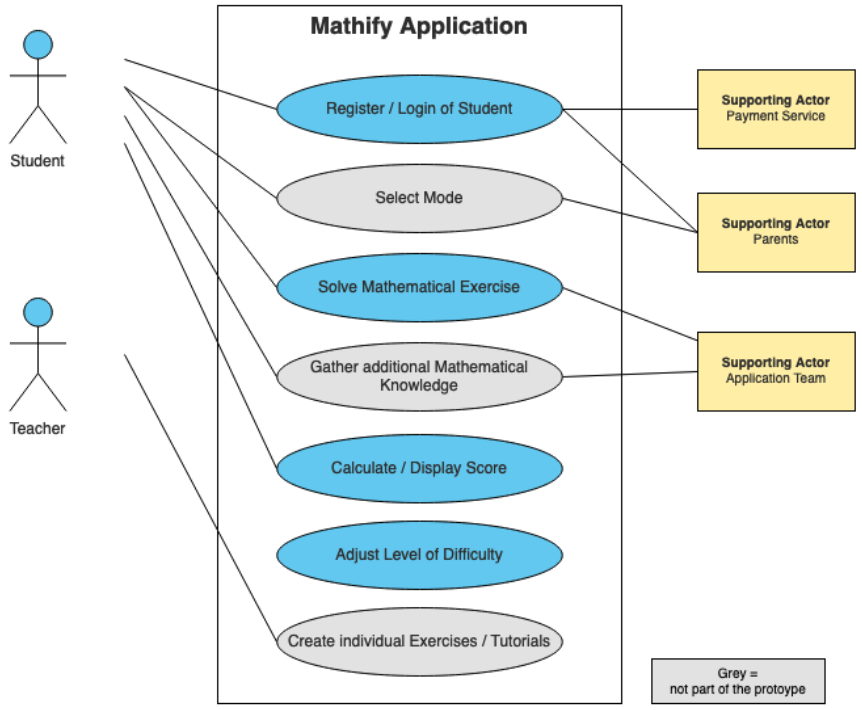

Mathify Application
Student
Teacher
Register / Login of Student
Select Mode
Solve Mathematical Exercise
Gather additional Mathematical Knowledge
Calculate / Display Score
Adjust Level of Difficulty
Create individual Exercises / Tutorials
Supporting Actor Payment Service
Supporting Actor Parents
Supporting Actor Application Team
Grey = not part of the protoype

Figure 1: Use-Case-Model

2.1.1 Register / Login of Student (Casual UC)

- Upon their initial login, a student, with or without parental assistance, is prompted to enter their username, password, grade, email address and payment details[16]. If the students already possesses an account, they are required to input their username and password. The system verifies all the information provided.
- The system offers a password reset option[17] for students who have forgotten their password, utilising the email address for assistance.
- Following successful login, the student selects a game mode, and the system identifies potential game statuses.
- The system then progresses to the exercise screen.

Figure 2 called "Mock-up register-screen" and figure 3 called "Mock-up login-screen" illustrate the registration and login interfaces. These mock-ups are conceptual representations designed to demonstrate the input fields and layout that a user will navigate during the initial access stages.

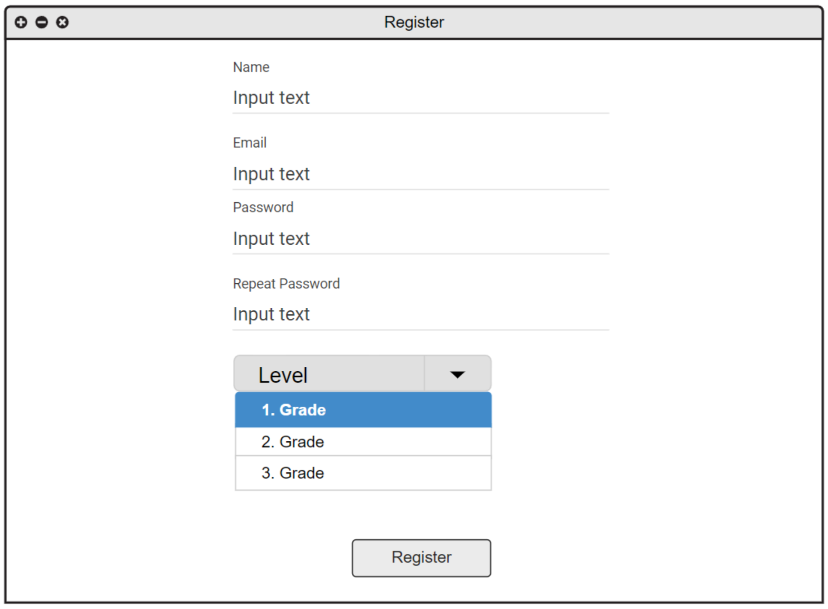


Figure 2: Mock-up Register-Screen

[16] Payment details are not part of the prototype.

[17] The password reset option is not part of the prototype.

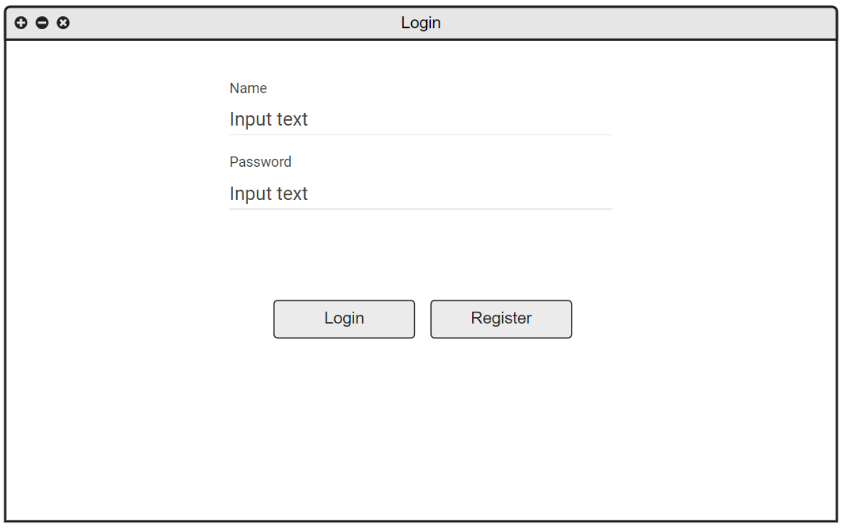


Figure 3: Mock-up Login-Screen

2.1.2 Select Mode (Brief UC)

- The student selects a mode[18]. The system determines the type of exercise and further functionalities (e.g. time-constraints) based on the selection of the mode.

Figure 4 called “Mock-up mode selection” showcases the interface for mode selection within the Mathify application. This mock-up visually details the options available—'Mixed mode' and 'Challenge'—illustrating the straightforward, user-friendly interface that guides students through their learning journey.

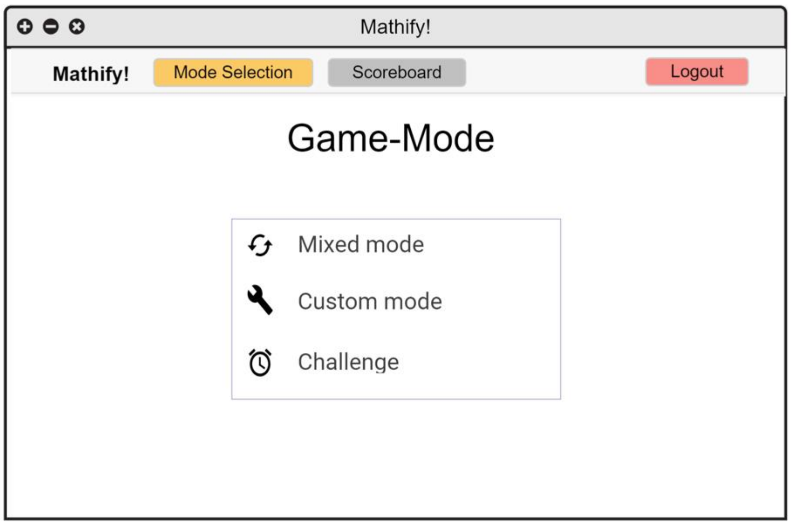


Figure 4: Mock-up Mode Selection

[18] Mode selection is not part of the prototype.

2.1.3 Solve Mathematical Exercise (Fully Dressed UC)

Given the complexity and detailed nature of the fully dressed use case it has been visualized in the following table 2 for clarity and ease of understanding.

Use Case Attribute	**Description**
Use Case Name	Solve Mathematical Exercise
Scope	Mathematics Quiz
Level	User Goal
Primary Actor	Student
Stakeholders and Interest	Teacher - Seeks to monitor the performance of their class. - Strives to identify and address any knowledge deficiencies. Parents - Aim to support their children's education. - Wish to keep track of their children's educational progress. Schools and Educational Department - Aim to raise educational standards across the nation. Tax Authority - Seeks to collect tax revenue.
Preconditions	The student must be properly logged in and licensed.
Success Guarantee / Postconditions	- Exercises are generated accurately. - Submissions are validated effectively. - Feedback is given back to the student. - Student's current progress (level + experience) is stored for future gameplay.
Standard Process	1. The system generates an exercise. 2. The student completes the exercise. 3. The system validates the student's submission. 4. The system verifies the correctness of the result. 5. The system provides feedback to the student. 6. The student repeats steps 1-5 to solve additional exercises.
Extensions / Alternate Flows	<u>If the student solves an exercise incorrectly:</u> 1. The system notifies the student of the incorrect result. 2. The student can try again. 3. The student can display the correct result by pushing a button.

	If the student's answer is in the wrong format (e.g. a word instead of a number): 1. The system generates an error message. 2. The system instructs the student on the correct format. 3. The student is prompted to re-enter their answer.
Special Requirements	1. Text must be legible from 0.5 meters away. 2. The system must process validations within 2 seconds. 3. The level of difficulty adjusts according to the correctness of the result.
List of Technical and Data Variations	- 2a): Input using a standard keyboard. - 2b): Input using an on-screen keyboard in the browser.
Frequency of Occurrence	- Whenever the student wishes to practice, which is the application's primary use case.
Open Questions	- How many exercise types from the syllabus will the program incorporate? - How to adjust the level of difficulty?

Table 2: Fully Dressed UC - Solve Mathematical Exercise

Figure 6 called "Mock-up mathematical quiz" illustrates the user-friendly quiz interface that students interact with. It displays a sample addition problem within the Mathify platform, demonstrating how questions are presented and how students input their answers. The interface includes helpful visual aids like a number line, enhancing the learning experience by providing contextual support that aids in problem comprehension and solution. This mock-up is a visual guide to the actual interface, highlighting features designed to make learning engaging and accessible for young users.

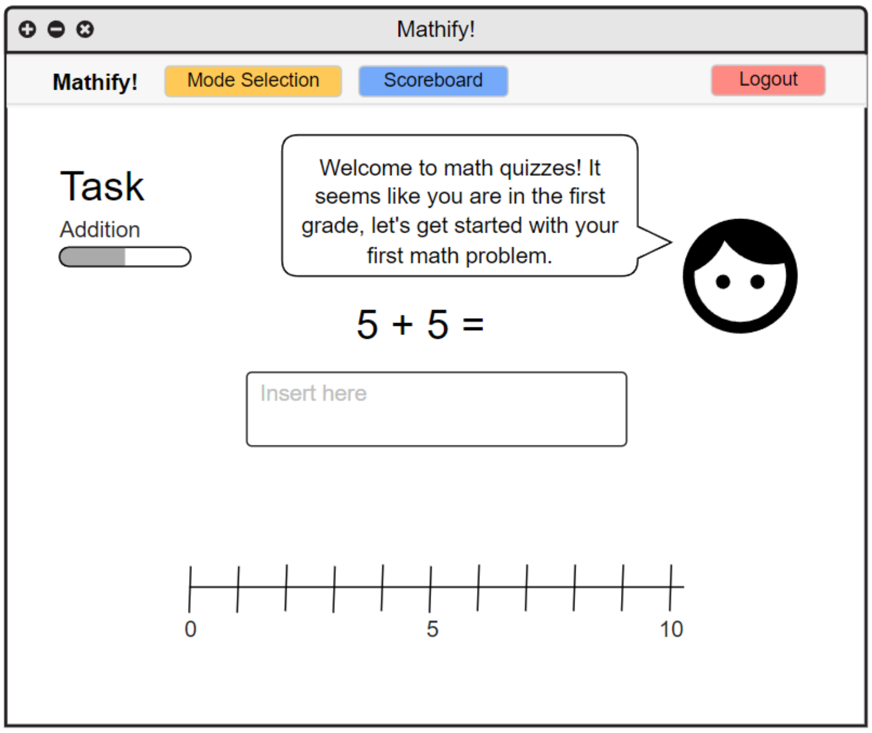

Mathify!
Mathify!
Mode Selection
Scoreboard
Logout
Task
Addition
Welcome to math quizzes! It seems like you are in the first grade, let's get started with your first math problem.
5 + 5 =
Insert here
0
5
10

Figure 5: Mock-up Mathematical Quiz

2.1.4 Gather additional Mathematics Knowledge (Casual UC)[19]

- The student can access supplementary material related to each exercise type from a comprehensive teaching library by pressing a button.
- Furthermore, when a low score is achieved on an exercise type, the system recommends additional learning materials automatically.
- Once the student feels confident with the topic, they return to the exercise screen to attempt further exercises.

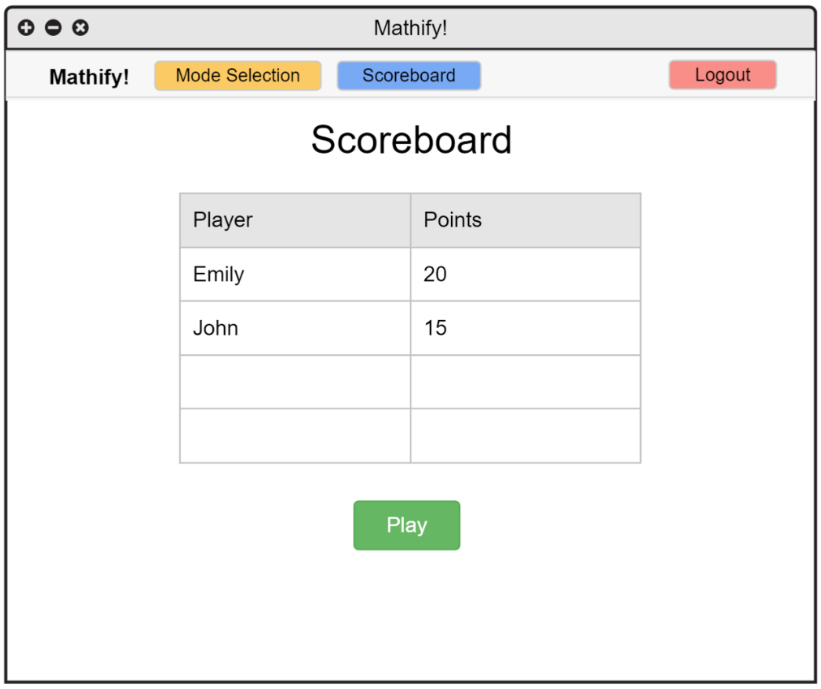
2.1.5 Calculate / Display Score (Brief UC)

- The system updates the student's score, awarding ten points for each correct answer and one point for a wrong one. Following each exercise, the scoreboard[20] entry is updated.
- Upon reaching a specified threshold, the scoreboard is displayed.

[19] This use case is not part of the prototype.

[20] See Glossary for a definition of this term.

As illustrated in figure 7, the mock-up scoreboard displays the system's dynamic update of scores, where points are allocated for both correct and incorrect answers, and the scoreboard is presented once a specified threshold is reached.



Player	Points
Emily	20
John	15

Figure 6: Mock-up Scoreboard

2.1.6 Adjust Level of Difficulty (Casual UC)

- When the student achieves a high score at a certain level of difficulty, a new level is unlocked, introducing more challenging exercises. The student can then choose to progress to the new level or opt to continue at the current level. After making their selection, the system reverts to exercise mode.
- Conversely, when the student scores low at a certain level of difficulty, the level is further lowered or remains the same if at the basic level already.

2.1.7 Create individual Exercises / Tutorials (Casual UC)[21]

- The teacher creates exercises or learning materials tailored to the class and uploads them to the application. The application checks the format and file size for compatibility.
- The teacher assigns the material to a specific topic and difficulty level or creates new topics altogether.
- The added content is then made available in the student's version of the application.

[21] This use case is not part of the prototype.

2.2 Additional Requirements

The following chapter describes additional functional and non-functional requirements not covered in the use cases. The requirements are formulated as user stories and are all measurable / verifiable. Additionally, the team has limited itself to only stating requirements that are necessary without providing specific solutions on how to achieve them.

<u>Our subject expert demands that...:</u>

1. the student requires a maximum of three clicks after logging in to reach the first exercise, as exceeding this number of clicks may overwhelm cognitive capacity.
2. a screen should never contain more than two full sentences, and a sentence should never exceed ten words, as experience shows that excessive reading quickly diminishes the playful component.
3. results are saveable to track learning progress together.
4. the application should have no downtime between 08.00 and 18.00 on weekdays to enable proper integration into the curriculum.

<u>Parents demand that...:</u>

1. the payment method is universally accepted, as no new payment medium is opened solely for using an application.
2. the learning progress is visible as they are only willing to pay for useful tools.

<u>Schools demand that...:</u>

1. changes in the national syllabus or new learning insights are continuously reflected in the application, as this is the only way a usage in the educational context makes sense.

<u>The government demands that...:</u>

1. data security must always be ensured because sensitive data[22] is stored in the system, which makes security necessary from a regulatory point of view.

[22] See Glossary for a definition of this term.

The company Mathify demands that…:

1. the application guarantees a traffic of 10,000 parallel users without breaking down to ensure credibility in the market and hence achieve profitability of the company.
2. The application must be compatible with common operating systems to quickly gain widespread adoption among students and push sales figures up.

2.3 Domain Model

The domain model[23] illustrated in figure 8 below represents Mathify through a simplified UML class diagram. It delineates the principal concepts with interconnection to each other to visualize their relationships. Attributes have been included where they enhance understanding.

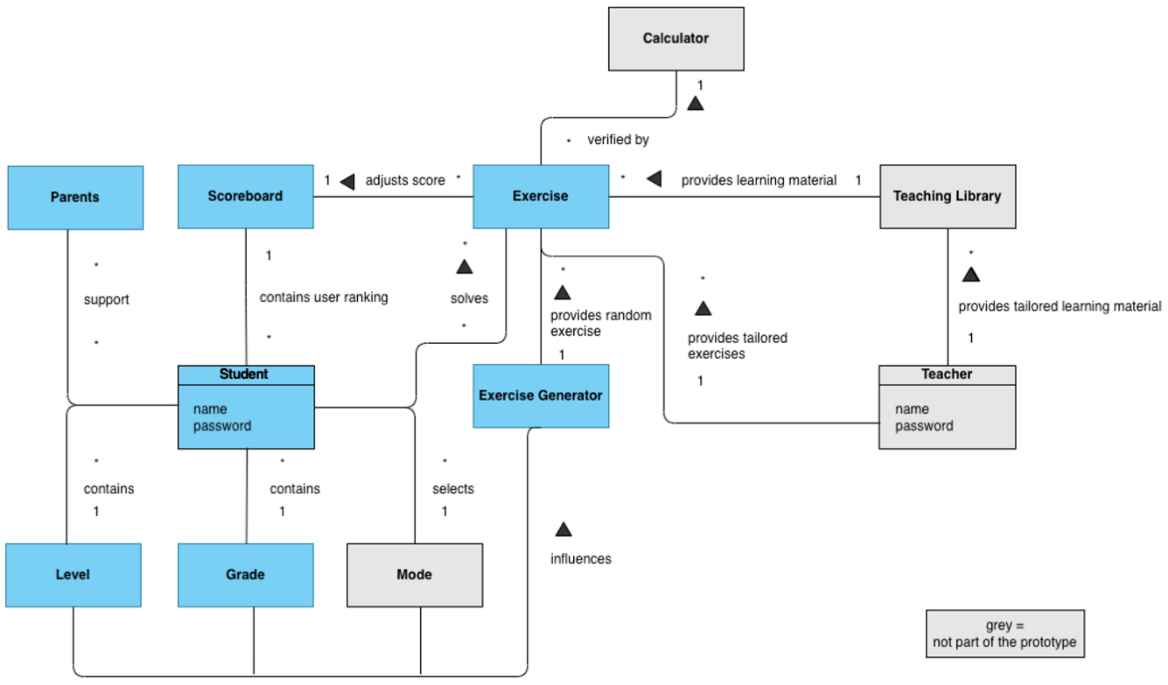


Figure 7: Domain Model

[23] See Glossary for a definition of the technical terms in the domain model.

3 Design

This chapter delves into the architectural design, choice of technologies, programming patterns, and interactions within the Mathify system. It illustrates how these elements combine to create a robust, maintainable, and scalable educational platform. The design phase is crucial as it lays the foundation for implementation, ensuring that the system's structure meets both the technical and business requirements effectively.

3.1 Software Architecture

The decision on a specific software architecture and technology stack is a critical step in the development of a project, which has far-reaching implications for its success, maintainability, and scalability. In the case of Mathify, the decision was made based on several factors such as team experience, specific project requirements and past experiences with technologies.

3.1.1 Programming Languages

Java was chosen as the primary programming language, mainly because of the existing experience within the team. This decision allows the team to be immediately productive, as no additional learning time for a new language is required. Java is also known for its robustness, performance, and wide support through a variety of libraries and frameworks, making it a safe choice for our backend development [9].

Python[24] will be introduced later for specialised mathematical calculations. Python offers an extensive selection of scientific libraries like NumPy [10] and SciPy [11], which are ideal for mathematical tasks. This decision enables us to leverage Python's capabilities in this area without having to build our entire architecture around it.

[24] Python is not part of the prototype.

3.1.2 Frameworks

Javalin, a lightweight framework for the backend, was chosen to simplify development and maximize performance [12]. Renowned for its simplicity and efficiency, Javalin enables backend development without the complexity and overhead associated with more extensive frameworks.

For the frontend, a combination of Angular [13], Bootstrap [14] and Angular Material UI [15] was selected. Angular allows for a modular and component-based architecture, promoting code reusability and a clear structuring of the application. Bootstrap and Angular Material UI complements Angular with simple and responsive design templates, accelerating the development of a professionally looking frontend.

3.1.3 Patterns

The use of the Factory Pattern supports the creation of the Exercise Generator by providing a clear separation between the creation of objects and their use. This enhances the flexibility and extensibility of the code. Moreover, polymorphism is integrated through an 'Exercise' interface, which is implemented by various exercise types. This approach ensures that the system maintains high cohesion and low coupling by allowing the same interface to be used for different underlying types of exercises, which can be extended anytime without affecting other parts of the codebase.

The Singleton Pattern ensures that only one instance of a class exists. The pattern is used with the UserRepository and the SessionHandler. The UserRepository reads the User.json on initialisation and loads all users into a list. This list should only live once per application. The SessionHandler handles the sessions for the users from the UserRepository. Due to the fact that the UserRepository only exists once, it is important to only have one session which is enforced by the singleton structure.

3.1.4 Architecture

The Model-View-Controller (MVC) Pattern [16] is employed to ensure a clear separation between the user interface, business logic, and data access layers. This separation facilitates ease of maintenance and scalability of the application.

The Client-Server architecture forms the basis of our system, enabling light and efficient communication between the frontend and backend, ensuring that the system can handle requests and scale as necessary without any significant changes to the client or server codebase.

3.1.5 Client-Server Architecture and Communication

In addition to the above-mentioned Client-Server architecture, it is important to detail the specific distribution model and communication strategy employed. Our application is a distributed system where the client (user's browser) interacts with our server through a web-based interface. The user accesses the website, which serves as the client, and makes requests to the server via an API. The server, built with our chosen tech stack, handles these requests in conjunction with user permissions, ensuring that users can only access the features and data they are authorised for. This is critical for maintaining security and data integrity across the system.

We opted for a RESTful API approach because it provides a standard and stateless communication protocol between the client and server, which simplifies scalability and the integration of future services [17]. This design choice supports our distribution model by allowing the client and server to evolve independently if the API contract is maintained. Furthermore, we chose to implement a straightforward and minimalistic approach to our system architecture to minimise the amount of boilerplate code necessary. This not only allows for quicker development and easier understanding of the system's workings but also reduces the potential for bugs and simplifies maintenance and future enhancements.

3.1.6 Reusable Logic and Modular Structures

Outsourcing the calculation logic into a separate maths package allows for its reuse in distinct parts of the application, improving consistency and reducing the likelihood of errors. Similarly, the "Security" package manages authentication processes, centralising security logic to ensure consistent and robust user verification across the application.

The Exercise Generator is an example of a modular structure accessible through the Factory Pattern. This ensures that changes to the creation logic are centralised, simplifying the maintenance, and testing of the application. This pattern is also applied to the Security package, where different authentication methods can be implemented or swapped with minimal impact on other components, thereby enhancing the system's flexibility and maintainability.

3.1.7 Package Diagram

The package diagram (Figure 9) provides a structured visualisation of the system's architecture, illustrating how different packages and their components are organised and interrelated within the Mathify application. This diagram is crucial for understanding the modular structure of the system and the dependencies between various software packages.

Overview

The package diagram divides the system into logical packages that encapsulate specific functionalities and responsibilities. These include:

Backend Packages: These are responsible for the core business logic and data management, and server-side functionalities. The main components are:

- Maths: Contains the logic for generating mathematical exercises and handling mathematical operations.
- Model: Manages data models and business rules.

Util: Handles all user-interface logic for the backend, ensuring that data is correctly passed to the frontend.

- Exercise: Manages exercise data and logic.
- Repository: Manages data storage and retrieval operations.
- Service: Provides backend services for data management and business operations.

API Packages: These include controllers and services that manage the API endpoints for client-server communication.

- Payment Service: A specialized package designed to handle payment processing and transaction management.
- Security: Ensures secure data handling and API access.
- Controller: Manages requests and routes them to the appropriate service layers.

UI Packages: Focuses on the frontend presentation and user interaction aspects. Key components include:

- Core: Includes essential UI components such as headers and authentication guards.
- Header: Provides user interface navigation and session management.
- Auth-Guards: Manages authentication and authorization processes.

Feature: Contains specific features like registration, scoreboard, and exercise views, each tailored to meet user interactions as defined by the system's requirements.

- Scoreboard View: Displays user scores and rankings.
- Tutorial View: Offers guided instructions and tutorials.

Exercise: It is designed to enhance user interaction and provide a seamless experience during exercises.

- Exercise View: Provides the interface for users to engage with exercises.
- Operations: Handles operational functionalities within the feature set.

Registration: Manages user sign-up and login processes.

- Grad-mode Selection: Allows users to choose their operational mode based on their grade level.
- User-Login: Manages user authentication.
- User-Registration: Handles new user sign-up processes.

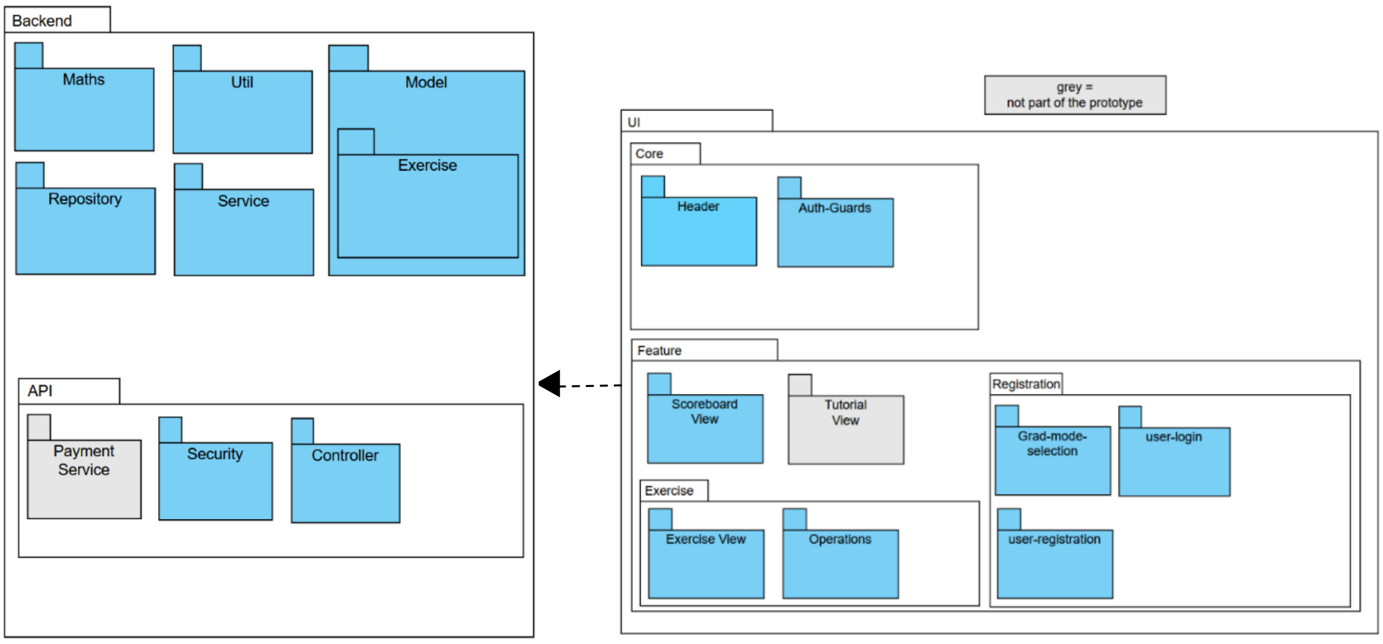


Figure 8: Package Diagram

3.2 Design Class Diagram (DCD)

The Design Class Diagram (DCD) for the Mathify system, which is visualised in figure 10 below, presents the structural relationships and interactions among its various components, including the User class, Grade, Exercise, and additional elements. The User class is tasked with handling user-related data and interactions, focusing on secure and personalised user experiences. Only key attributes and methods are listed, which include managing users, authenticating, and customising user experiences.

The Exercise interface, implemented in the MathsExercise class, offers a variety of mathematical challenges suitable for students of different grade levels. This mirrors the system's educational aims. The MathsExercise class includes essential methods such as verifying results and transforming DTOs.

The Grade enumeration is crucial in monitoring student performance and development. It interacts with the Exercise classes to evaluate and record students' achievements in the exercises they undertake.

Classes such as MathsGenerator and ExerciseSubType are vital for creating exercises that suit the student's grade level and preferred types of mathematical exercises, ensuring the content is fit for each student's learning stage. Only pivotal methods have been listed to maintain clarity and focus.

The Router class is key in managing API calls, directing requests to the correct controllers, such as the UserApiController and ExerciseApiController. This routing system is central to the data flow and provides endpoints for the frontend to interact with the backend services.

Supporting classes like UserRepository and Settings play significant roles in the persistence of user data and application configuration, respectively. The UserRepository is involved in the storage and management of user data, while Settings manage application parameters, including security configurations such as HTTPS. These descriptions include only the principal methods to illustrate the system's functionality.

The DCD highlights the mechanisms of data processing and representation within the Mathify system, illustrating a design focused on maintaining an effective educational platform.

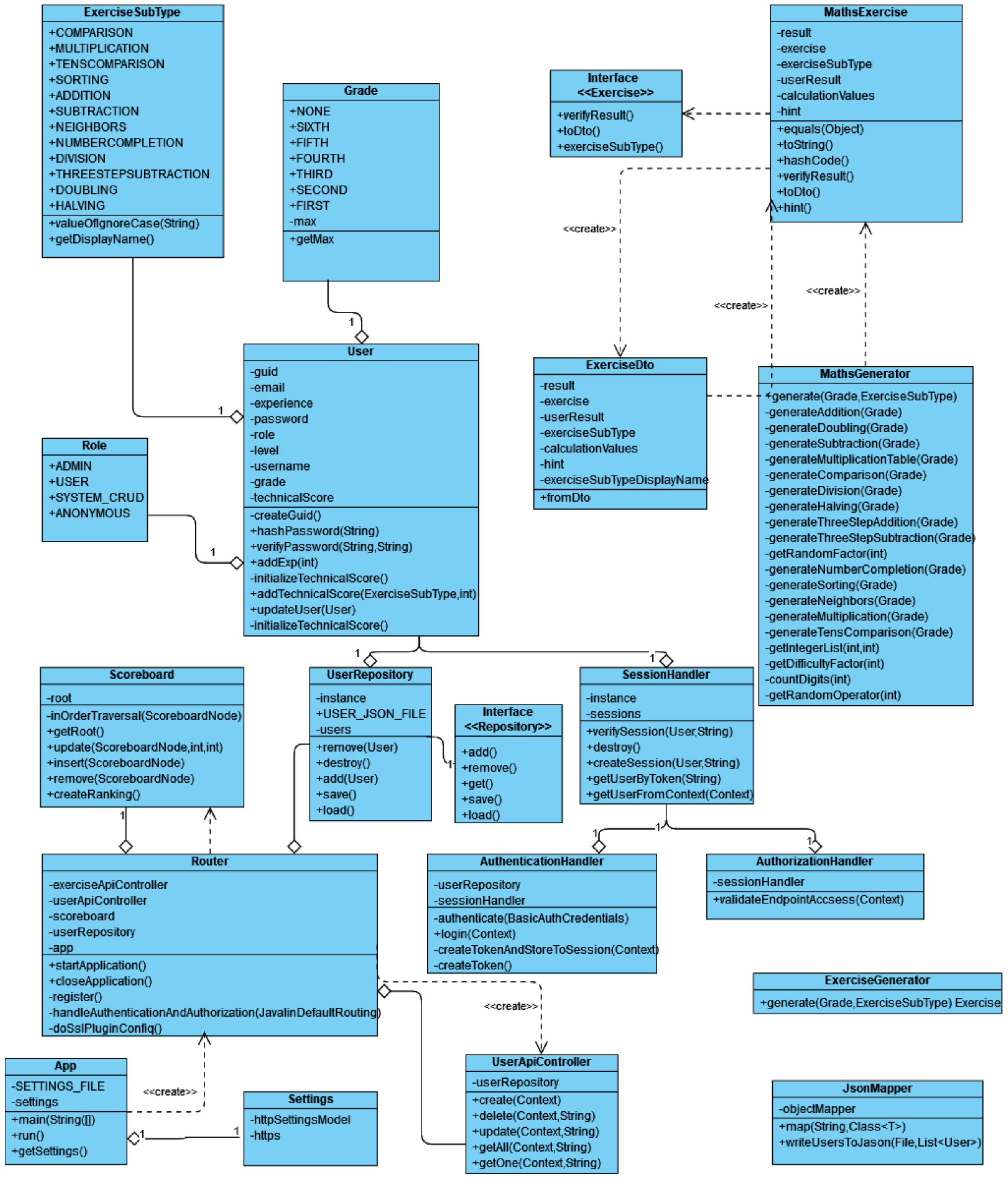


Figure 9: Design-Class Diagram

3.3 Interaction Diagram

Exercise Generation

The exercise generation process, which is visualised below in figure 11, is a critical component of the Mathify system. Students request exercises from the system, which then generates tasks tailored to their level of experience. This generation process continues until the student decides to stop. This dynamic approach helps keep the exercises relevant and challenging, appropriate to the student's progressing skill level.

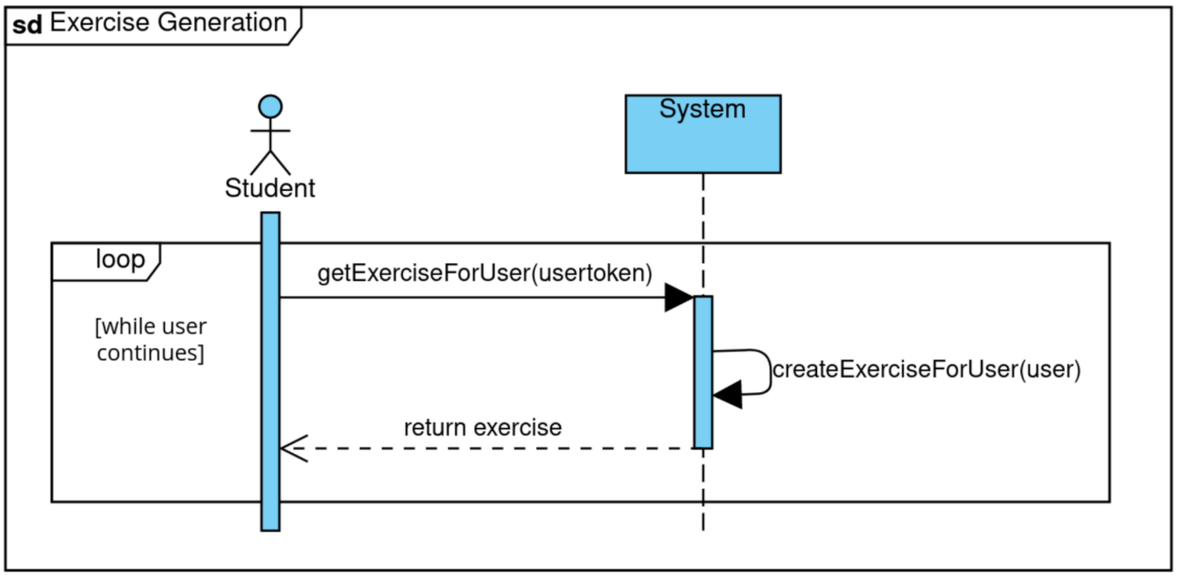


Figure 10: Exercise Generation

3.4 Registration Process

This section maps out the journey of a new user as they join the Mathify community. The registration process, which is visualised below in figure 12, involves the student entering their unique username, a secure password, their email address, and their current grade. The system checks if the submitted information is valid and if the username or email already exists. If the data is unique and valid, the user is successfully created; otherwise, the system generates an error message, informing the students of the need to adjust their registration details.

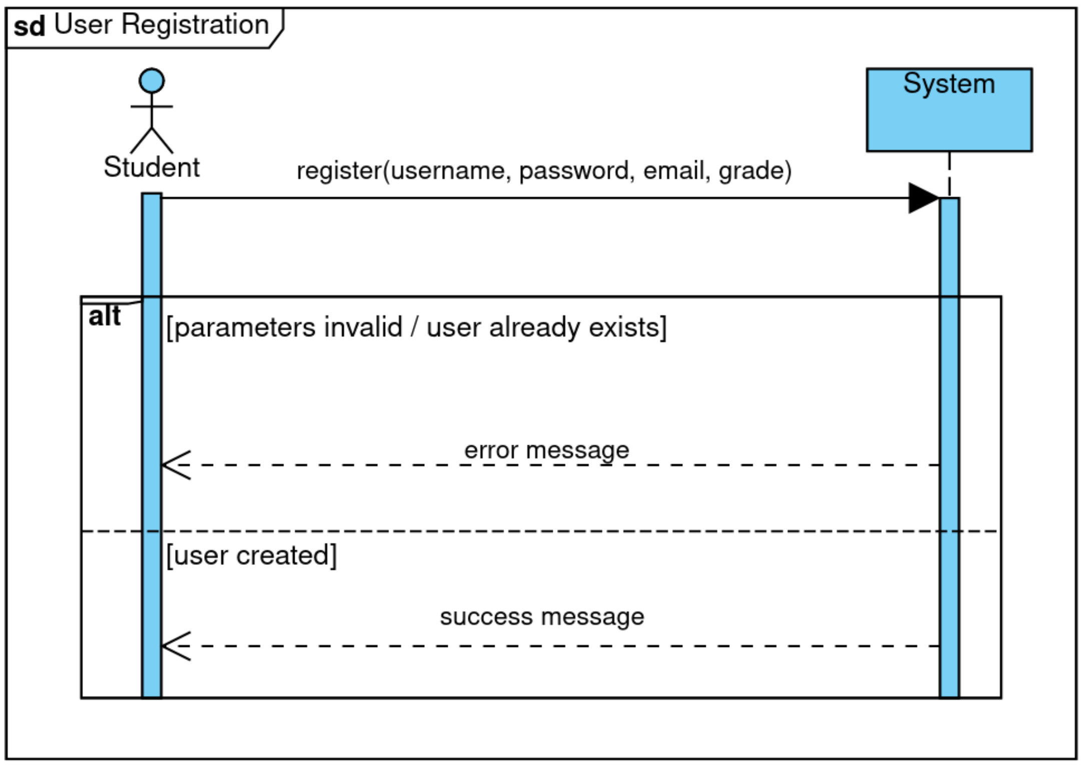

sd User Registration
Student
System
register(username, password, email, grade)
alt
[parameters invalid / user already exists]
error message
[user created]
success message

Figure 11: Registration Process

Score Calculation

This chapter outlines the validation sequence that occurs after a student completes an exercise. As illustrated in figure 13 below, the system verifies the result of the exercise; if the answer is correct, the student is awarded 10 experience points. If incorrect, they receive just 1 point. Additionally, the student receives feedback regarding the accuracy of their response. This process ensures that students understand their performance and learn from their mistakes.

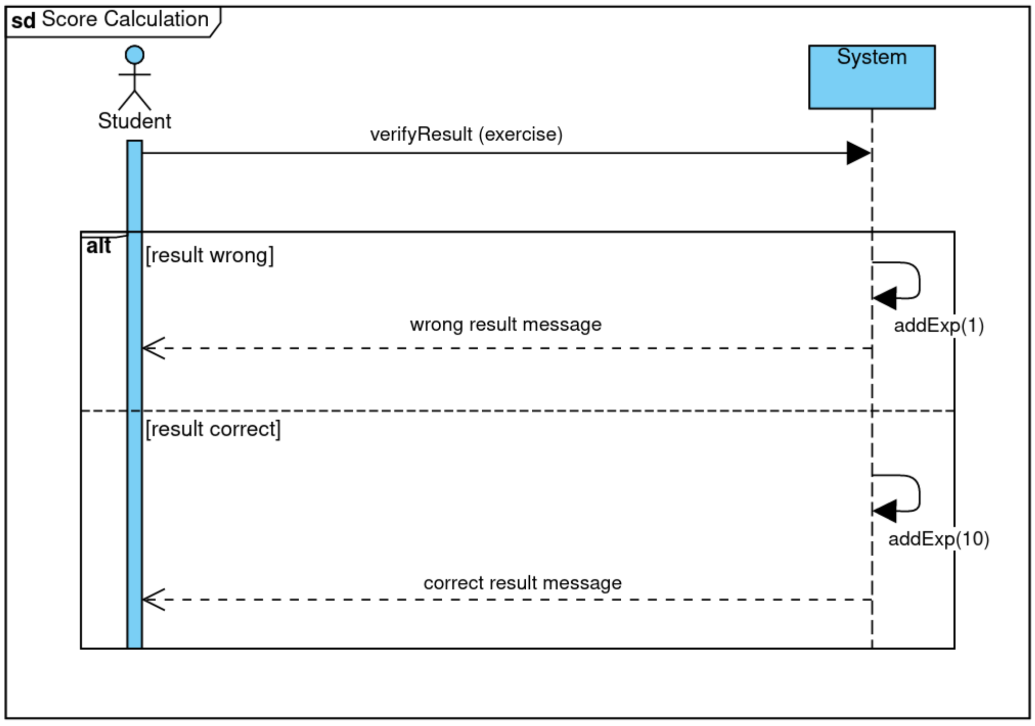


Figure 12: Score Calculation

Adjust Difficulty

Adjust Difficulty," depicted in Figure 13, illustrates the system's adaptive learning feature. It personalises the difficulty level of exercises based on the accuracy of a student's responses. After a student attempts an exercise, their technical score is adjusted accordingly. If the response is correct, the score is increased, otherwise, it is decreased. This adjustment is designed to be independent of the user's overall experience level, ensuring that each student's learning path is tailored to their specific needs.

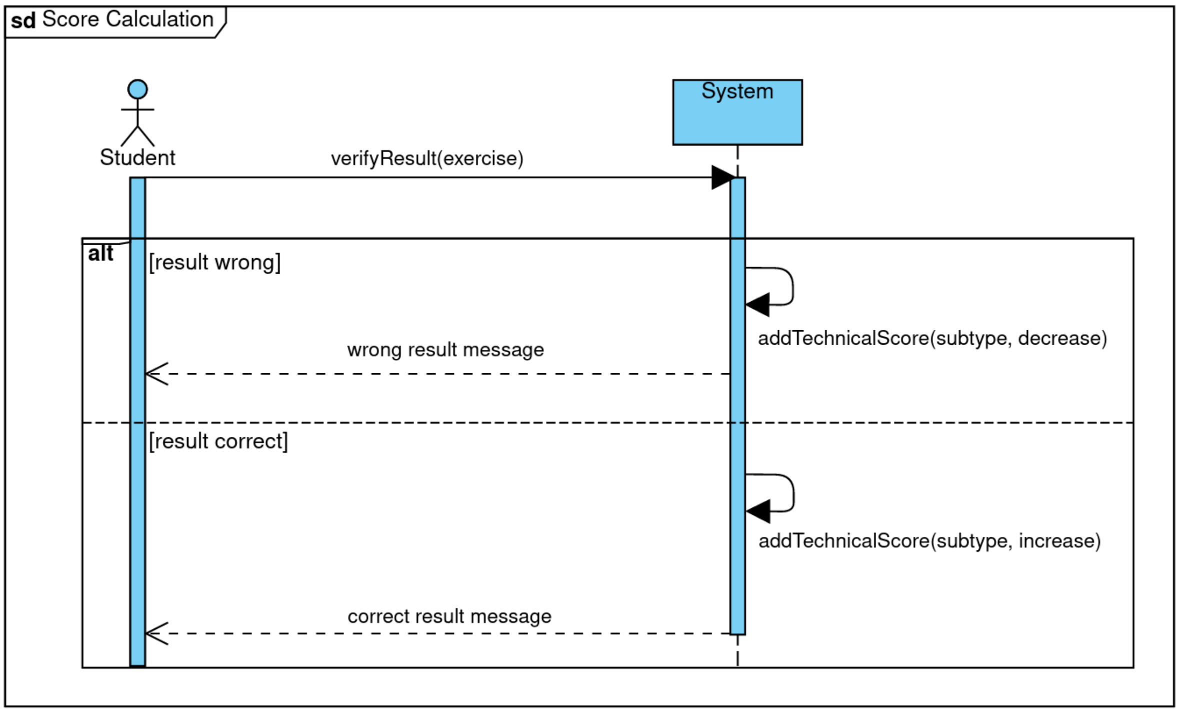


Figure 13: Adjust Difficulty

4 Implementation

This chapter outlines the delivery of Mathify's software components and the rigorous testing procedures employed to assure quality and functionality.

4.1 Deliverables

The tangible outcomes of the development process are manifold, encompassing both backend and frontend deliverables. The backend is neatly packaged through a Docker setup [18], as detailed in the accompanying Readme file, consolidating the application's services for seamless deployment and operation. The frontend, crafted to provide an intuitive and responsive user experience, will be accessible via www.mathify.ch.

4.2 Testing Concept

The rigour of the testing paradigm is multifaceted, involving a comprehensive blend of unit, black-box, and white-box testing to underpin the application's reliability. Continuous testing is increasingly important for efficient software delivery. Effective testing helps prevent bugs, improve performance, and ensure security, ultimately leading to higher-quality software products and greater customer satisfaction [19].

4.2.1 Testing Strategy

A spectrum of tests was developed to scrutinise every aspect of Mathify, including:

- Unit Tests: Empirical and straightforward, these tests validate the core functionality of each unit within the system.
- Integration Tests: Evaluate the harmonious operation of interconnected components within Mathify, such as the seamless collaboration between the frontend and backend.
- System Tests: Entail a complete boot-up of the application followed by a range of REST requests to emulate actual usage scenarios and validate the integrated system's performance.

4.2.2 Testing Tools

A suite of tools was utilised to perform these tests with precision and efficiency: JUnit and Mockito for mock and stub testing, and REST Assured for API testing.

4.2.3 Testing Environment

The development environment, particularly IntelliJ, was where most tests were created and executed.

4.3 Overview of All Tests

Comprehensive testing was conducted across different components of the application to ensure robustness and reliability. Below is a detailed compendium of the test classes and methodologies:

4.3.1 Unit Tests

API Controller Tests:

- ExerciseApiControllerTest: Conducts unit tests to validate the functionalities of exercise-related API endpoints. These tests ensure that the API responds.
- UserApiControllerTest: Carries out unit tests to ensure the correct functioning of user-related API endpoints, focusing on CRUD operations and user authentication flows.

Security Tests:

- AuthenticationHandlerTest: Involves unit tests for verifying authentication mechanisms, such as password hashing and token generation.
- AuthorisationHandlerTest: Executes unit tests to check the system's ability to correctly grant or deny access based on user roles, ensuring secure access control.
- SessionHandlerTest: Unit tests to ensure that sessions are managed securely, including session creation, expiration, and renewal mechanisms.

Math Tests:

- MathsGeneratorTest: Unit tests to confirm that the exercise generation algorithm produces the expected mathematical challenges, assessing the accuracy and variability of the outputs.

Model Tests:

- ScoreboardTest: Unit tests that assess the logic behind the scoreboard's data processing and presentation, ensuring that scores are calculated and displayed correctly.
- UserTest: Conducts unit tests on the User model's functionality including data validation, persistence, and lifecycle management.

Utility Tests:

- JsonMapperTest: Unit tests for verifying the accurate mapping of data structures to and from JSON format. These tests ensure that data sent to and received from the frontend is correctly serialised and deserialised.

4.3.2 Integration Test

Router Tests:

- RouterTest: Integration tests that evaluate the routing logic, ensuring that HTTP requests are correctly interpreted and directed to the appropriate handlers.

Service Tests:

- ExerciseServiceTest Integration tests that examine services related to exercise data manipulation and logic application, ensuring that services interact correctly with the database and other application layers.

4.3.3 System Test

Frontend Integration Tests:

- These tests are manually performed to validate the complete path from request to response on the frontend. Integration tests here involve real user scenarios to ensure that the frontend interface is intuitive, responsive, and aligns seamlessly with backend operations.

This comprehensive suite of tests ensures that each component of Mathify not only meets the required specifications but also interacts seamlessly with others, providing a secure, dependable, and user-friendly experience.

5 Results

The original project idea for Mathify was to create a dynamic and engaging mathematics quiz platform for students in grades 1 through 9, making mathematics fun through a game-like experience where students could progress through levels, earn points, and receive badges at their own pace. The application aimed to reduce mathematics anxiety by celebrating effort and perseverance, fostering a positive and confidence-boosting environment for both struggling and enthusiastic learners. It was also designed to provide parents and teachers with a supportive tool to enhance a student's mathematical skills and passion for learning.

5.1 Summary of Achieved Objectives

From the project's conception, it was understood that the time frame would only permit the realisation of a Minimum Viable Product (MVP). This strategic approach enabled the team to prioritise core functionalities that would embody the essence of Mathify. The objectives for the MVP that were successfully met are:

- Crafting of an intuitive and engaging graphical user interface.
- Implementation of a fully functional login system.
- Class selection interface that adapts exercises to the chosen grade, reflecting the Lehrplan 21 syllabus in each exercise.
- Development of exercises for grades 1 to 3, offering a variety of challenges suited to each learning stage.
- Creation of a gamified experience completed with scoreboards and motivational feedback, deeply integrated with the Lehrplan 21 to ensure educational relevance.

5.2 Open Points

Initial plans did include exercises for grades 4 to 6, but continuous evaluation highlighted the necessity to concentrate efforts on developing a polished and fully operational platform for grades 1 to 3. This focus was essential to ensure the product's quality and manageability.

While the main objectives have been met without outstanding issues, it is important to acknowledge that the existing features could potentially be further enhanced. To be candid, the graphical user interface could be significantly more engaging. Furthermore, the motivational feedback provided could be more varied and inspiring. Thus, a notable area for improvement would be to enhance the overall aesthetics and user engagement features of the platform. It is essential, however, to remember that this platform is still in its prototype phase.

5.3 Outlook for Possible Further Developments

Payment Service - Necessity for Revenue Generation

To make Mathify economically viable and to fund ongoing development, the integration of a reliable payment service is crucial. This will facilitate the transition from a free to a premium model, allowing for the monetization of advanced features.

Password Reset Option - Essential for User Convenience

An automated password reset option is a standard feature for modern applications, reducing the overhead of manual support and enhancing user experience.

Various Modes - Enriching User Engagement

Diverse game modes, including time-boxed challenges and competitive modes against other students, are critical for a gamified learning experience that keeps students returning and practicing.

Badge System Motivation through Recognition

Implementing a badge system can significantly boost motivation. The comparison and collection of badges or stickers could serve as a strong incentive for ongoing student participation and engagement.

Tailored Features for Teachers - Utility for Educators

To make Mathify a tool of choice in educational institutions, features allowing teachers to create and feed in custom exercises and mock exams are essential. This functionality could make the application indispensable for classroom use.

Tutorial Library - Knowledge Acquisition and Support

A comprehensive tutorial library is vital to assist students beyond exercise solving. It can help students understand underlying concepts, thereby enriching the learning process and making the application more appealing to parents and teachers.

In-Application Purchases and Additional Services - Extended Monetization Strategy

Future considerations could include features such as access to expert tutors via the application for personalised guidance or customisable avatars with special accessories that can be purchased, adding a layer of personal touch to the user experience.

Conclusion - Sustaining Innovation and Relevance

The above proposals are the planned features for Mathify. However, creativity and innovation are necessary to keep pace with evolving educational and technological landscapes. Keeping features aligned with educational trends and user expectations will be key to Mathify's long-term success and relevance. Therefore, more features beyond the list are likely over the years to come.

6 Lists & Figures

6.1 Bibliography

[1] U. I. for Statistics. (2017, September). *Fact Sheet No. 46 - More Than One-Half of Children and Adolescents Are Not Learning Worldwide* [Online]. Available: https://uis.unesco.org/sites/default/files/documents/fs46-more-than-half-children-not-learning-en-2017.pdf. [Accessed: 28.04.2024].

[2] OECD. (2023). *PISA 2022 Results (Volume I): The State of Learning and Equity in Education* [Online]. Available: https://doi.org/10.1787/53f23881-en. [Accessed: 28.04.2024]

[3] OECD. (2023). *PISA 2022 Results (Volume II): Learning During - and From - Disruption* [Online]. Available: https://doi.org/10.1787/a97db61c-en. [Accessed: 28.04.2024]

[4] OECD. (2023). *PISA 2022 Results - Factsheet Switzerland* [Online]. Available: https://www.oecd.org/publication/pisa-2022-results/webbooks/dynamic/pisa-country-notes/95f719cc/pdf/switzerland.pdf. [Accessed: 28.04.2024]

[5] A. B. Erzinger, G. Pham, O. Prosperi, and M. Salbisberg. (2023). *PISA 2022 - Die Schweiz im Fokus* [Online]. Available: https://www.sbfi.admin.ch/dam/sbfi/de/dokumente/2023/12/pisa2022_nationaler_bericht.pdf. [Accessed: 28.04.2024]

[6] solocode GmbH. *Various pages* [Online]. Available: https://anton.app/de/. [Accessed: 28.04.2024]

[7] Khan Academy. *Various pages* [Online]. Available: https://de.khanacademy.org. [Accessed: 28.04.2024]

[8] Prodigy Education. *Various pages* [Online]. Available: https://www.prodigygame.com/main-en/. [Accessed: 28.04.2024]

[9] Oracle. (2023). *Introduction to Java* [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html. [Accessed: 11.05.2024]

[10] N. Developers. (2023). *NumPy: The fundamental package for scientific computing with Python* [Online]. Available: https://numpy.org/. [Accessed: 11.05.2024]

[11] P. S. Foundation. (2024). *SciPy: Open Source Scientific Tools for Python* [Online]. Available: https://www.scipy.org/. [Accessed: 11.05.2024]

[12] Javalin. (2024). *Javalin - A simple web framework for Java and Kotlin* [Online]. Available: https://javalin.io/. [Accessed: 11.05.2024]

[13] Angular. (2023). *Introduction to Angular* [Online]. Available: https://angular.io/guide/what-is-angular. [Accessed: 11.05.2024]

[14] Bootstrap Organization. (2024). *Bootstrap Documentation* [Online]. Available: https://getbootstrap.com/docs/5.1/getting-started/introduction. [Accessed: 11.05.2024]

[15] Angular Material UI Organization. (2024). *Angular Material UI Component Library* [Online]. Available: https://material.angular.io/. [Accessed: 11.05.2024]

[16] Mozilla Developer Network. (2024). *Model-View-Controller (MVC)* [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/MVC. [Accessed: 11.05.2024]

[17] Mozilla Developer Network. (2024). *REST* [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/REST. [Accessed: 11.05.2024]

[18] Docker Inc. (2024). *Docker Documentation* [Online]. Available: https://docs.docker.com/. [Accessed: 11.05.2024]

[19] IBM Corporation. (2024). *Software Testing* [Online]. Available: https://www.ibm.com/topics/software-testing. [Accessed: 11.05.2024]

[20] Bundesamt für Statistik. (2023). *Obligatorische Schule* [Online]. Available: https://www.bfs.admin.ch/bfs/de/home/statistiken/bildung-wissenschaft/personen-ausbildung/obligatorische-schule.html. [Accessed: 28.04.2024]

6.2 List of Figures

6.3 List of Tables

6.4 Glossary

Term	Definition / Explanation
Application-Team	The collective responsible for the development and maintenance of the Mathify quiz platform.
Calculator	A component within the application to assess and validate the accuracy of answers submitted by the student.
Exercise	A mathematical challenge tailored to the student's grade, selected mode, and reached level of difficulty.
Exercise Generator	A system component that creates a variety of random mathematical exercises for students.
Game Mode	An initial selection feature that allows students to choose a play style such as mixed or custom.
Grade	A categorization that ranges from 1 to 9, dictating the types of exercises generated based on educational level.
Teaching Library	A repository of educational resources, including videos, texts, and graphs to support learning.
Level of Difficulty	A progression parameter within the game that progresses from basic to intermediate and finally to advanced.
Mathify	The official name of the quiz application.
National Syllabus	The "Lehrplan 21" dictates the variety and complexity of exercises tailored to each grade and level of difficulty.
OECD Countries	38 countries that are members of the Organisation for Economic Co-operation and Development.
Parents	Caregivers responsible for nurturing and supporting the educational development of their children
Scoreboard	A feature that enables students to compare their performance with peers and their learning journey.

Sensitive Data	All personal information that can identify clients, requiring secure storage to protect privacy.
STEM	STEM is an acronym that stands for Science, Technology, Engineering, and Mathematics.
Student	The primary user group for the Mathify quiz, encompassing students from grades 1 through 9.

Table 3: Glossary

7 Annex

7.1 Project Management

This chapter outlines the project management details for the development of Mathify, structured through a series of sprints designed to achieve sequential milestones. Each sprint focuses on specific functionalities and improvements, with efforts gauged aga inst planned objectives.

7.1.1 Sprint 1

The initial sprint of the Mathify project focused on establishing the foundational elements of the application. The scope of this sprint, along with the planned and actual efforts, is detailed in Table 4.

Task	Planned Effort	Actual Effort	Result
Establishing the initial project structure	S	S	Completed
Modelling user and exercises	M	M	Completed
Setting up back- and frontend	M	L	Partial Completion
Building a backend testing environment	M	M	Completed
Formulating a skill table as a basis for generating exercises	S	M	Completed
Compiling a readme document	S	S	Completed

Table 4: Sprint 1 Review

The primary goal not achieved in this sprint was the implementation of backend authentication, which has been deferred to Sprint 2. This postponement was necessary due to the task requiring more effort than initially anticipated, along with the team's limited experience in this area. Additionally, the effort estimate for the 'Skill Table' was adjusted from 'S' to 'M', indicating an underestimation of the complexity involved in understanding the national syllabus, 'Lehrplan 21,' and in identifying the most effective types of exercises for each grade to optimize educational outcomes. Despite these challenges, the project's schedule buffer is deemed sufficient to absorb these delays, and as such, no immediate corrective actions have been taken.

7.1.2 Sprint 2

In the second sprint, the team focused on addressing deferred tasks and introducing new functionalities. The progress and objectives of this sprint are detailed in Table 5.

Task	Planned Effort	Actual Effort	Result
Implementing backend authentication	M	M	Completed
Implementing GUIs for user registration, grade selection and mode selection	M	L	Registration Window Pending
Testing frontend to backend connection	M	L	Pending Implementation
Creating good draft for technical report 1	M	M	In Progress
Implementing grade 1 exercises	S	M	Completed
Implementing backend scoreboard	L	S	Completed

Table 5: Sprint 2 Review

Although the GUI for user registration and the connection between the frontend and backend have not been established yet, the team remains optimistic. The morale and confidence of the team are high, and there is a unanimous belief that the final goals will be met without the need for additional measures. The project is on track to proceed as planned with strong ongoing collaboration among team members.

7.1.3 Sprint 3

During the compressed timeline of one week in Sprint 3, the team focused on crucial integrations and basic user interface, as illustrated in table 6. This sprint was critical in achieving seamless integration between the frontend and backend systems, and in significantly refining the exercise generation process to enhance user interaction and responsiveness.

Task	Planned Effort	Actual Effort	Result
Connecting front- and backend	M	L	Completed
Creating login page	M	M	Completed
Refactoring exercise generation	S	S	Completed
Implementing grade 2 exercises	S	S	Completed
Creating router mock tests and improving existing ones	M	M	Completed
Creating window for registration	S	S	Completed
Implementing scoreboard and exercise presentation in frontend	M	M	Partial Completion
Finalising technical report 1	M	L	Completed

Table 6: Sprint 3 Review

Sprint 3 successfully met most of its objectives with the seamless integration of system components and the introduction of a functional login page. However, the implementation of the scoreboard and exercise presentation in the frontend was only partially completed. These components demonstrated the need for further refinement to meet the project's standards for content and usability. Moreover, seemed the technical complexity higher than initially expected.

7.1.4 Sprint 4

Sprint 4 concentrated on enhancing the user experience and the technical robustness of the Mathify platform. The team worked on introducing a more secure, token-based user authentication system and adjusting the exercise difficulty algorithm to better respond to user performance, as visible in table 7.

Task	Planned Effort	Actual Effort	Result
Changing exercises for easier testing	M	M	Completed
Introducing technical score to user	M	M	Completed
Replacing user credentials with token	L	L	Completed
Adjusting exercise difficulty	S	M	Completed
Moving exercise selection from front- to backend	M	M	Completed
Implementing grade 1-3 exercises in frontend	M	L	Partially completed
Implementing grade 1-3 exercises in backend	M	L	Partially completed
Adding “Zahlenstrahl” to view	S	S	Completed
Verifying result in frontend	M	S	Completed
Showing scoreboard in frontend	S	M	Completed
Creating a good draft for technical report II	L	L	In Progress

Table 7: Sprint 4 Planning

The efforts to enhance security and user engagement were largely successful, with the new token-based authentication system and updated exercise difficulty settings fully implemented. However, the sprint faced challenges in fully completing the implementation of grades 1-3 exercises in both the frontend and backend. Particularly one exercise type could not be completed due to much higher complexity than initially expected. This partial completion indicated a need for additional resources and focus in the subsequent sprint to ensure these core functionalities are polished and fully operational.

7.1.5 Sprint 5

In Sprint 5, visualised in table 8 below, the focus was on finalizing the core functionalities for grades 1 to 3 and on polishing the overall system. The team concentrated on backend optimization to enhance performance and on frontend enhancements to provide a more engaging and intuitive user interface. Additional support features, such as comprehensive exercise hints, were incorporated to facilitate student learning.

Task	**Planned Effort**	**Actual Effort**	**Results**
Finishing grade 3 exercise in backend	M	L	Completed
Creating helper methods in backend for frontend	M	M	Completed
Adding hint texts for exercises in backend	S	S	Completed
Cleaning up backend	M	M	Completed
Improving user experience in frontend	M	L	Ongoing
Implementing multiplication table in frontend	M	M	Completed
Fixing bugs	M	M	Completed
Finalising technical report II	L	L	Completed

Table 8: Sprint 5 Planning

Sprint 5 was successful in completing most of its goals, significantly improving the backend's efficiency and the frontend's user interaction. The grade 3 exercises were fully implemented in the backend, and substantial enhancements were made to assist user navigation and learning in the frontend. However, some aspects of the user experience improvements remain ongoing. These continuous optimizations are scheduled to be resolved in future updates, ensuring that Mathify continues to evolve to meet user needs and expectations effectively.

7.2 Updated Risk Assessment

This chapter outlines the risk management strategies and its evolution from initial assessment to their status as of May 12, 2024.

Risk Assessment and Management

As illustrated in figure 15, several key risks were identified at the project's inception, with their probability and severity re-evaluated over time as follows:

1. Technical Complexity

Initially assessed with high probability, severity, and overall risk, the technical complexity risk has seen significant mitigation. This reduction in probability from high to low is attributed to the team's increasing familiarity with the project's complexities and the acquisition of technical expertise. The high impact potential remains due to the foundational nature of this risk, yet effective planning and adaptability have lessened its likelihood.

2. Software Bugs & Glitches

Maintaining a medium probability and severity, the risk of software bugs and glitches has been managed through comprehensive testing. Although the mitigation strategies have substantially decreased the incidence of critical bugs, the inherent nature of software development maintains a residual risk at a constant level.

3. User Interface & Experience (UI/UX) Mistakes

The risk associated with UI/UX mistakes remains high in probability due to the fact that the project team has no access to a large group of students as pilot user. The severity, classified as low, reflects the team's ability to make iterative adjustments to the interface based on user feedback and testing, thereby managing potential negative impacts on user satisfaction and engagement.

4. Compliance with Educational Standards

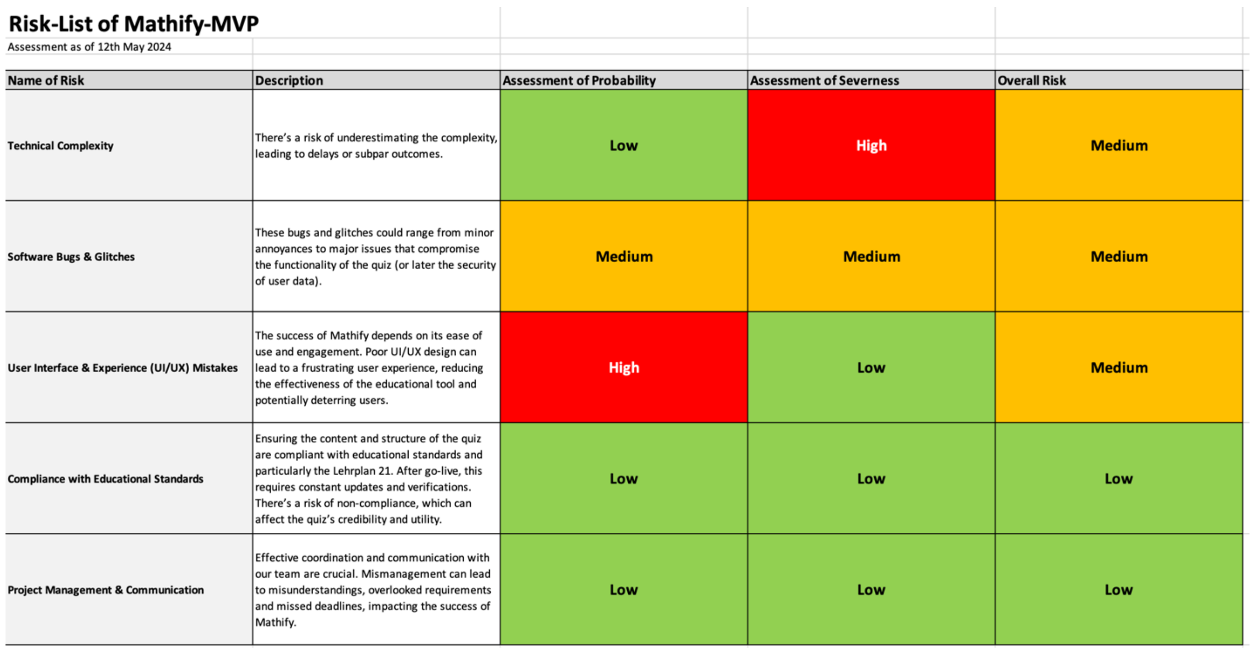
Initially rated with a medium probability, the risk related to compliance with educational standards such as Lehrplan 21 has been successfully reduced to low through expert involvement. This expert oversight has ensured that the quiz content remains aligned with educational mandates, effectively lowering the probability of non-compliance risks.

5. Project Management & Communication

Originally assessed with medium probability, severity, and overall risk, this category has seen significant improvement. Through effective teamwork and enhanced communication strategies, the probability and impact of project management and communication issues have been minimized to low. The collaborative culture within the team has fostered a reliable and resilient project management structure.

Conclusion

The Mathify project has demonstrated substantial progress in risk management, with significant reductions in both the likelihood and impact of initially high-risk categories. The continued focus on adaptive strategies, expert consultation, and rigorous testing protocols has been integral to this success.



Risk-List of Mathify-MVP

Assessment as of 12th May 2024

Name of Risk	Description	Assessment of Probability	Assessment of Severness	Overall Risk
Technical Complexity	There's a risk of underestimating the complexity, leading to delays or subpar outcomes.	Low	High	Medium
Software Bugs & Glitches	These bugs and glitches could range from minor annoyances to major issues that compromise the functionality of the quiz (or later the security of user data).	Medium	Medium	Medium
User Interface & Experience (UI/UX) Mistakes	The success of Mathify depends on its ease of use and engagement. Poor UI/UX design can lead to a frustrating user experience, reducing the effectiveness of the educational tool and potentially deterring users.	High	Low	Medium
Compliance with Educational Standards	Ensuring the content and structure of the quiz are compliant with educational standards and particularly the Lehrplan 21. After go-live, this requires constant updates and verifications. There's a risk of non-compliance, which can affect the quiz's credibility and utility.	Low	Low	Low
Project Management & Communication	Effective coordination and communication with our team are crucial. Mismanagement can lead to misunderstandings, overlooked requirements and missed deadlines, impacting the success of Mathify.	Low	Low	Low

Figure 14: Risk-List

7.3 Exercise Groundwork

The table displayed below was developed in collaboration with a senior teacher to serve as the foundation for implementing exercises in our quiz application. It categorizes the mathematical skills and exercises intended for students from 1st to 3rd grade, organized into three levels of difficulty: Basic, Intermediate, and Advanced. Each column corresponds to a grade level, outlining the progression of skills that students are expected to acquire—from simple additions and subtractions to more complex tasks such as multiplication and division with three-digit numbers. This structured approach ensures that the curriculum is precisely tailored to the needs of each student group.

Grade	1st Grade	2nd Grade	3rd Grade
Difficulty			
Basic	**Find neigboring numbers (1 to 5)** Example 1) 2 - 3 - ? --> 4 2) 2 - ? - 4 --> 3 3) ? - 4 - 5 --> 3	It is usual to review and deepen the advanced material from the previous class (i.e. advanced section in this file) at the beginning of the new school year.	It is usual to review and deepen the advanced material from the previous class (i.e. advanced section in this file) at the beginning of the new school year.
	Find the right sequence (1 to 5) Example 1) 5, 3, 4 --> 3 - 4 - 5 2) 1, 3, 2 --> 1 - 2 - 3	**Doubling and halving numbers (1 to 10)** Example: 1) Double the number 4 2) Half the number 8	**Addition / subtraction up to 1000, with entire 100s)** Example: 1) 200 + 300 2) 700 - 200
	Compare numbers (1 to 5) Example 1) Find the biggest number: 3, 1, 4 --> 4	**Three-step addition / subtraction (1 to 10)** Example: 1) 2 + 3 + 5 2) 5 - 2 - 1	**Find neighboring numbers (up to max. 1000)** Example: 1) 154 - 155 - ?
Intermediate	**Addition (1 to 10)** Example: 1) 5 + 3 2) 1 + 4	**Two-digit addition (result max. 100)** Example: 1) 23 + 32	**Rounding numbers to tens (up to max. 100)** Example: 1) 22 --> 20 2) 15 --> 20
	Subtraction (1 to 10) Example: 1) 4 - 2 2) 7 - 3	**Two-digit subtraction (subtrahend = number of tens)** Example: 1) 54 - 20	**Addition / subtraction with 100s and 10s** Example: 1) 180 + 130 2) 600 - 120
	Complete missing number (1 to 10) Example: 1) 4 + ? = 8 --> 4		
Advanced	**Addition / subtraction / completion (1 to 20)** Example: 1) 7 + 6 2) 15 - 3 3) 10 + ? = 15	**Two-digit subtraction** Example: 1) 78 - 25	**Written addition (up to max 1000)** Example: 1) 528 + 201
	Addition / subtraction / completion (number of tens) Example 1) 20 + 30 2) 40 - 10 3) 60 + ? = 70	**Single-digit multiplication** Example: 1) 5 * 3	**Written addition with carrying over (up to max. 1000)** Example: 1) 98 + 24
	Comparison of number of tens (10 to 100) Example: 1) Which number is bigger: 20 or 50?	**Multiplication table (numbers 1 to 9 - ascending)** Example: 1) 1 * 9 2) 2 * 9 ... (until 10 * 9)	**Written multiplication with three-digit numbers** Example: 1) 4 * 178
		Division without remainder (1 to 20) Example: 1) 20 / 4	**Written division with three-digit numbers (without and with remainder)** Example: 1) 280 / 20 2) 290 / 20
			Order of operations (German: "Punkt-vor-Strich") (single-digit) Example: 1) 7 + 5 * 2
Key Competency	Number line (German: Zahlenstrahl)	Simple multiplication & division	Written calculation without number line

Table 9: Exercise Logic Grade 1 to 3

7.4 Full Business Case

Creating a business case involves analysing several key factors such as market need, financial projections, competitive positioning, and potential risks and mitigations. Here is an outline of the business case:

Step 1: Market Analysis (in very short, cf. previous chapters)

- Target Audience: The primary focus encompasses the community of approximately one million students [20] enrolled in Swiss schools, which includes a mix of public and private institutions.
- Market Need: A customised learning tool that aligns with the national syllabus of Switzerland, enhancing engagement through gamification.
- Competition: Identified three direct competitors and how Mathify can differentiate itself from them.[25]

Step 2: Product Offering & Pricing

Basic Subscription: "Math Explorer": Designed for individual students needing basic practice. The low price point is attractive for casual learners.

- Features: Access to a basic set of exercises.
- Content Updates: No monthly updates.
- Support: No access to online help but only general FAQs.
- Pricing: CHF 1 per month or CHF 10 per year for one login.

Premium Subscription: "Math Adventure": Enhanced features and support are designed to meet the needs of dedicated learners and parents deeply committed to their children's educational success.

- Features: Comprehensive access to all exercise types and modes, including leader boards, score saving and badges.
- Content Updates: Regular monthly updates with new quizzes and challenges.
- Support: Includes access to online help and FAQs.
- Pricing: CHF 5 per month or CHF 50 per year for one login. Includes a free trial period of one month to encourage sign-ups.

[25] cf. chapter 1.4 "Competition Analysis" for more details.

School Edition: “Math Champions”: This package offers extensive features and support tailored for classroom use, ideal for schools seeking comprehensive digital learning solutions.

- Features: Multi-user licenses suitable for classroom use, incorporating all features found in the “Math Adventure” subscription.
- Professional Development: Resources and webinars for teachers to effectively integrate the application into their curriculum.
- Customisation: Options to tailor content to classroom needs and the capability for teachers to create automatic or customized mock exams.
- Content Updates: Monthly updates with new materials designed to enhance classroom learning.
- Support: Comprehensive online support, plus additional technical assistance for initial setup and orientation in schools.
- School Engagement: Exclusive contests and challenges designed to foster healthy competition among classes or schools.
- Pricing: CHF 30 per year per student, with a minimum annual subscription fee of CHF 600. Includes a free 1-month demo and pilot period.

Step 3: Financial Projections

- Revenue Streams: Subscriptions (Basic, Premium, School Edition), possibly augmented by in-application purchases or additional services.[26]
- Pricing Strategy: Competitive pricing with free trials to boost early adoption.
- Development Costs: Mainly salary costs, initially set at CHF 20,000 per year as the team develops almost for free as an investment in the application. However, these costs are anticipated to rise gradually, reaching CHF 100,000 per year. This increment is attributed to the recruitment of more skilled programmers who are necessary to enhance the sophistication and capabilities of the application as it evolves.
- Five-Year Projections[27]:
  - Year 1: Focus on application development and pilot testing in selected schools. Limited or no revenue expected.
  - Year 2: Launch publicly, expect to reach 100 Basic, 1,000 Premium and 400 School Edition subscriptions.

[26] The decision on whether to offer in-application purchases or additional services has not yet been made. This aspect is currently under evaluation as part of our business planning process.

[27] All calculations are based on average annual payments, simplifying the payment model by combining monthly and yearly subscriber fees.

- Year 3: Expansion to more schools, marketing push with costs up to CHF 250,000 aiming for 200 Basic, 5,000 Premium and 2,000 School Edition subscriptions.
- Year 4: Consolidate presence, refine offerings based on feedback, targeting 300 Basic, 10,000 Premium and 6,000 School Edition subscriptions.
- Year 5: Sustained growth phase, potentially exploring partnerships or expansions, with goals of 500 Basic, 20,000 Premium and 15,000 School Edition subscriptions.

Step 4: Marketing and Sales Strategy

- Launch Phase: Starting with CHF 50,000 yearly, mainly utilising digital marketing, partnerships with schools, and word-of-mouth in educational communities.
- Growth Phase: Steadily increasing to CHF 300'000 yearly with a focus on credible testimonials, case studies, and demonstrable learning outcomes to attract new users.
- Retention Strategies: Keeping costs at CHF 300'000 per year by regular updates, engagement features, and responsive customer service to maintain user interest and satisfaction.

Step 5: Operational Plan

- Development: Agile development cycle with periodic updates and feature rollouts. The initial technology implementation will be kept cost efficient, with an annual budget of CHF 10,000 per year.
- Roll-out and Support: Implementation of more advanced technology, hiring of new people for multi-tier support system based on subscription level, ensuring appropriate resource allocation, all creating total run rate costs up to CHF 200'000 per year.

Step 6: Business Plan Risks

- Adoption Risks: Risk can be mitigated by demonstrating the educational value of the product and ensuring alignment with the national syllabus of Switzerland.
- Competition Risks: Stay competitive by continuously innovating and enhancing the user experience.
- Financial Risks: Adopt a conservative budget strategy and scale operations according to revenue growth to ensure financial stability (cf. table below).

Step 7: Milestones

- Development Milestones: Beta version by year 1, full launch by year 2.
- Financial Milestones: Almost break-even by year 3, substantial profits by year 4.