DEVELOPMENT OF AN AUTOMATIC CODE GRADING PLATFORM

BY

PERVOLARAKIS MICHAIL

BSc Thesis, Informatics Engineering, Hellenic Mediterranean University, 2022

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2022

Approved by:

Associate Professor

Nikolaos Vidakis

ABSTRACT

Technology plays a big role in everyone's everyday life in the twenty-first century and has altered many procedures that would otherwise be done manually or even be impossible without the assistance of technology. Computers have played an important role in technological growth since they are the instruments that people use to improve their quality of life. Entertainment, work, and even education have all profited significantly from technological advancement.

Education has undergone a profound transition as a result of the technology revolution. The use of computers to distribute learning-related content that leads to the development of new knowledge and skills is referred to as E-learning. Nowadays, E-learning is utilized practically everywhere since it is more efficient because it can be tailored to each student and can also be less expensive because it can reach thousands of people whereas a classroom can only hold a limited number of pupils. Asynchronous online courses, synchronous online classrooms, and other forms of E-learning exist.

One of the results of technological advancement is the creation of several opportunities for both technology-related enterprises and IT professionals. In truth, many of the Fortune 500 companies are tech firms, while the bulk of the non-tech firms on the list use technology in a variety of ways. This, combined with the fact that younger generations are accustomed to engaging with technology on a daily basis, has resulted in a high demand for IT-related education.

This thesis discusses the creation of an E-learning web platform called Eurytus. Eurytus is a platform that aims to assist in fast and efficient programming learning. Professors can give their students programming homework using Eurytus. These assignments may (a) include input and output tests to determine whether the algorithms perform as expected, (b) structure checking to determine whether the submitted code adheres to the professor's defined structure, and (c) design pattern checking to determine whether the submitted code complies with the professor's requested design patterns. Students can utilize the web app's IDE to complete the programming tasks, and the platform will evaluate their submitted code automatically in accordance with the guidelines laid out by the instructor. The platform also offers the ability for individual users to create and publish programming challenges for everyone to join, that way community members can create and join exams created by others in order to check and improve their programming skills.

Cutting-edge technologies like react JS, express JS, and mongo DB were used to create the Eurytus platform. The platform was built using a microservices architecture for added reliability, and an event-based system was used to facilitate communication between each service. This along with the use of docker and Kubernetes for deployment makes the app more reliable and safer because all the services are independent of one another and can continue to function even if one of them fails.

ΠΕΡΙΛΗΨΗ

Στον 21ο αιώνα η τεχνολογία παίζει τεράστιο ρόλο στην καθημερινότητα όλων μας, αλλάζοντας κάποιες διεργασίες οι οποίες παλαιότερα θα έπρεπε να γίνουν χειροκίνητα ή ακόμα θα ήταν αδύνατο να πραγματοποιηθούν χωρίς την χρήση της τεχνολογίας. Οι υπολογιστές έπαιξαν μεγάλο ρόλο στην τεχνολογική ανάπτυξη καθώς είναι τα εργαλεία που χρησιμοποιούνται από τους ανθρώπους για την αύξηση του βιοτικού τους επιπέδου. Τομείς όπως η διασκέδαση, η εργασία και η εκπαίδευση έχουν επωφεληθεί ιδιαίτερα από την τεχνολογική ανάπτυξη.

Η τεχνολογική επανάσταση έχει επιφέρει τεράστιες αλλαγές στην εκπαίδευση. Η χρήση υπολογιστών για την διανομή εκπαιδευτικού υλικού που οδηγεί στην απόκτηση γνώσης και ικανοτήτων ονομάζεται E-learning. Στις μέρες μας το E-learning χρησιμοποιείται σχεδόν παντού καθώς είναι πιο αποτελεσματικό γιατί μπορεί να διαμορφωθεί στις ανάγκες των μαθητών και πιο φθηνό καθώς μπορεί να απευθυνθεί σε ένα τεράστιο κοινό ενώ οι συμβατικές τάξεις μπορούν να χωρέσουν συγκεκριμένο αριθμό μαθητών. Ασύγχρονα διαδικτυακά μαθήματα καθώς και διαδικτυακές τάξεις αποτελούν κάποιες από τις μορφές του E-learning.

Ένα από τα αποτελέσματα της τεχνολογικής εξέλιξης είναι η δημιουργία αρκετών ευκαιριών για εταιρίες που ασχολούνται με την τεχνολογία καθώς και άτομα που δουλεύουν στον τομέα της πληροφορικής. Για την ακρίβεια πολλές από τις μεγαλύτερες εταιρείες ασχολούνται με την τεχνολογία η χρησιμοποιούν την τεχνολογία με έναν ή περισσότερους τρόπους. Το γεγονός αυτό μαζί με το γεγονός ότι οι νέοι μεγαλώνουν με την τεχνολογία ως μέρος της καθημερινότητας τους έχει οδηγήσει σε αυξημένη ζήτηση σε εκπαίδευση στην πληροφορική.

Ο σκοπός της παρούσας εργασίας είναι η δημιουργία μιας E-learning πλατφόρμας που ονομάζεται Eurytus. Η συγκεκριμένη πλατφόρμα στοχεύει να βοηθήσει στην εκμάθηση προγραμματισμού. Μέσω της πλατφόρμας οι καθηγητές μπορούν να αναθέσουν ασκήσεις προγραμματισμού στους μαθητές τους. Αυτές οι ασκήσεις μπορούν να περιέχουν τεστ τα οποία ελέγχουν (α) αν η έξοδος του αλγορίθμου για κάποια συγκεκριμένη είσοδο είναι η αναμενόμενη, (β) αν η δομή του αλγορίθμου ταιριάζει με την δομή που έχει ορίσει ο καθηγητής και (γ) αν ο αλγόριθμος περιέχει τα design patterns που θέλει ο καθηγητής. Οι μαθητές μπορούν να χρησιμοποιήσουν έναν IDE για να λύσουν την προγραμματιστική άσκηση και να την υποβάλουν σε αυτόματη βαθμολόγηση σύμφωνα με τις οδηγίες βαθμολόγησης του καθηγητή. Επιπλέον, η πλατφόρμα προσφέρει την δυνατότητα σε μέλη της κοινότητας να δημιουργήσουν ασκήσεις οι οποίες αφού ελεγχθούν από τους

διαχειριστές δημοσιεύονται δίνοντας την δυνατότητα σε οποιονδήποτε να δοκιμάσει τις ασκήσεις για να βελτιώσει η να αποκτήσει προγραμματιστικές γνώσεις.
Για την ανάπτυξη της πλατφόρμας Eurytus χρησιμοποιήθηκαν σύγχρονες τεχνολογίες όπως react JS, express JS και mongo DB. Επιπλέον η πλατφόρμα υλοποιήθηκε με αρχιτεκτονική micro-services και η επικοινωνία μεταξύ των services έγινε με χρήση events. Αυτό μαζί με την χρήση τεχνολογιών όπως Kubernetes και docker κάνουν την πλατφόρμα πιο ασφαλή και αξιόπιστη καθώς όλα τα services λειτουργούν ανεξάρτητα το ένα με το άλλο με αποτέλεσμα αν ένα από αυτά δεν είναι διαθέσιμο τα άλλα δεν επηρεάζονται και συνεχίζουν να λειτουργούν κανονικά.

List of contents

ABSTRACT ii

ΠΕΡΙΛΗΨΗ iv

List of contents vi

List of figures ix

List of tables xi

1 Introduction 1

1.1 Objectives 2

1.2 Outline 2

2 Background 3

2.1 E-learning 3

2.2 Code Testing 5

2.3 Code structure and design 8

2.4 Microservices 12

3 Background work 16

3.1 Topcoder.com 16

3.2 Coderbyte.com 16

3.3 Codewars.com 17

3.4 Codingame.com 17

3.5 Hackerrank.com 18

3.6 Leetcode.com 18

4 Software Design 20

4.1 Problem Description 20

4.2 Software Analysis 21

4.2.1 Platform’s scenarios of use 21

4.3 Eurytus Design 26
4.3.1.1 Architecture 26
4.3.1.2 Component diagram 27
4.3.2 Eurytus’ s functionalities, procedures, and events 31
5 Software Implementation 37
5.1 Technologies used 37
5.1.1 Docker 37
5.1.2 Kubernetes 39
5.1.3 React JS 40
5.1.4 JavaScript & Typescript 41
5.1.5 Node JS 41
5.1.6 Mongo dB 42
5.1.7 Express JS 42
5.1.8 HTML & CSS 43
5.2 Software implementation 43
5.2.1 Execute challenges implementation 43
5.2.2 Challenge moderation implementation 48
5.2.3 Authentication implementation 52
5.2.4 Event communication implementation 54
6 Results / demo 56
6.1 Auth page demo 56
6.2 Challenges page demo 57
6.3 Solve challenge page demo 57
6.4 Create challenge page demo 58
6.5 User profile page demo 61
6.6 Challenge statistics page 62
6.7 Admin page demo 63

7 Conclusion & future work 65

REFERENCES 66

List of figures

Figure 1: Microservices architecture [52] 13
Figure 2: High level Architecture Diagram 27
Figure 3: Eurytus components 28
Figure 4: Eurytus common components 29
Figure 5: Execute challenges service components 30
Figure 6: Eurytus UI components 31
Figure 7: Authentication flow diagram 32
Figure 8: Challenge creation flow diagram 33
Figure 9: Edit challenge flow diagram 34
Figure 10: Delete challenge flow diagram 35
Figure 11: Solve challenge flow diagram 36
Figure 12: Request moderation flow diagram 36
Figure 13: Vms vs Containers [76] 38
Figure 14: Docker architecture [77] 38
Figure 15: Pseudocode of output testing 44
Figure 16: Class structure comparison function 45
Figure 17: execute challenge sequence 46
Figure 18: Execute challenges performance before and after optimization 47
Figure 19: Challenge operation sequence diagram 49
Figure 20: Challenge moderation sequence diagram 50
Figure 21: Approving an update request 51
Figure 22: Pending requests stack 52
Figure 23: Get current user method as implemented inside Eurytus common module 53
Figure 24: Version checking when receiving an update challenge event 55
Figure 25: Authentication page demo 56
Figure 26: Challenges page demo 57
Figure 27: Solve challenge page demo 58
Figure 28: Challenge description demo 59
Figure 29: Challenge test creation demo 61
Figure 30: User profile page demo 62
Figure 31: Challenge statistics page 63

Figure 32: Admin page demo..64
Figure 33: Review pending request demo..64

List of tables

Table 1: Comparison of similar platforms 19

1 Introduction

The twenty-first century is distinguished by accelerated and rapid technical innovation. It has evolved and influenced practically every discipline of research as well as every aspect of everyone's daily life. Computers have played a significant part in the advancement of technology since they are the tools that people utilize to improve their quality of life. The term computer applies not just to personal computers, laptops, or smart devices that people use to access the internet, but also to televisions, cars, trains, and other systems that rely on some form of computer or computer chip to function properly.

There has been a dramatic transformation in education over the years of the technological revolution. Traditional educational systems have evolved to keep up with technological advances, pushing professionals, educators, and students to reevaluate their fundamental beliefs in order to use technology to re-design or re-engineer the education and training system [1]. E-learning is described as the use of internet technology to deliver learning-related material that leads to the development of new knowledge and skills [1], [2]. Today's kids are referred to as digital natives, having been born in a technologically advanced era, and as a result, education must evolve to keep up with them. Learning Management Systems, Massive Open Online Courses and other forms of eLearning exist. Despite variances in eLearning styles and, in some circumstances, platforms, they all share fundamental ideas and, in some situations, approaches.

The rapid advancement of technology has prompted businesses to use technology to perform previously manual operations. There is also a surge in the number of tech companies, which are businesses that deal with digital electronics and internet-based services. In fact, many of the Fortune 500 companies are tech companies, while the majority of the non-tech companies on the list employ technology in a variety of ways. As a result, IT workers are in high demand. This, combined with the fact that younger people grow up with technology as part of their daily life, has resulted in an increase in computer science degree enrollments.

Despite the demand, however, there are insufficient resources to learn or enhance programming abilities. Online coding learning platforms [3]–[8] are a well-known and cost-effective choice. Coding challenge platforms are a subset of the larger subject of education and programming training, and they are extremely important in computer science. These platforms offer a variety of coding challenges that aim to help develop and improve programming skills [9].

In light of this, a platform for creating and grading programming assignments is proposed. Professors can create programming assignments for their students to complete and grade them using input output tests, structural checks, and even design pattern checks. People who desire to learn or improve their programming skills can use it by engaging in community-created and maintained challenges.

1.1 Objectives

As previously stated, the goal of this thesis is to create an automated code grading platform. Using the suggested system, professors will be able to design and construct coding assignments and examinations for their students. Learners will then be able to complete these assignments utilizing an IDE running on the platform. When participants have completed their assignment, they can turn it in. The system will automatically grade these solutions by running the tests that the instructors have provided. The same automatic code grading technique will be used to develop a feature for creating and maintaining public challenges. This enables the community to create and maintain challenges in which everyone can participate, making the platform ideal not only for educators but also for anyone looking to challenge themselves with programming challenges in order to learn new skills or improve existing ones via practice.

1.2 Outline

This thesis is organized as follows: In chapter 2, an extensive description of the theoretical and technological background of this work will be presented, while in chapter 3, a brief overview of applications similar to the one presented in this thesis will be presented. The platform's design and key goals are outlined in chapter 4. Following the design, chapter 5 presents a detailed look on how the previously designed modules were implemented which can also be used as a documentation of the developed services. The final outcomes of this thesis are provided in a brief demo in chapter 6. Finally, in chapter 7, the last ideas and future work directions are reflected.

2 Background

This section of the thesis will offer the theoretical and technological foundation required to completely understand the thesis's contents. This section, in particular, discusses e-learning and how its use affects today's education, as well as code testing and how it might aid programmers. Following that, there is a discussion on code structure, how it impacts the final program, and how to create good structure. Finally, this section covers information on microservices, including the benefits of adopting microservices and the challenges of implementing a microservices-based architecture.

2.1 E-learning

Technology plays an increasingly important part in our everyday lives in the twenty-first century, prompting professionals, educators, and students to reconsider their fundamental ideas in order to use technology to re-design or re-engineer the education and training system [1]. E-learning is often described as the use of internet technologies to deliver information related to the learning feature that builds new knowledge and skills [1], [2]. E-learning can be used as a substitute for traditional education or as a supplement to it [1]. E-learning can be synchronous or asynchronous, and it can be provided via the internet, interactive multimedia, educational games/simulations, and social media [2], [9], [10]. Since younger students are comfortable with new technology and are possibly drawn by new sorts of learning methods, E-Learning is considered as one of the fastest expanding trends in education and is becoming increasingly popular in institutions of higher learning, such as universities [9], [11].

Types of E-learning

E-learning can be classified into several categories based on how it is provided. The two main types of e-learning are synchronous E-learning and asynchronous E-learning. Both learning styles have their pros and cons, and the technique that is best for a learner varies widely.

Asynchronous E-learning

Asynchronous E-learning does not require students and lecturers to be present in real time online [10]. Usually, the study material has been prepared in advance and students can complete it at their own pace [9], [10]. Even though asynchronous e-learning allows students to regulate their own time and learning, it is recommended that online courses be scheduled, which means that there should be a deadline for students to finish their assignments [12]. Web blogs, streaming video, streaming audio are all examples of asynchronous e-learning [2], [9].

Synchronous E-learning

Synchronous e-learning necessitates the participation of students and professors at many places at the same time. It refers to any real-time learning event that incorporates instantaneous two-way contact between participants and is presented to distant learners [9], [10]. Even though synchronous E-learning brings students together, it might be advantageous to introverts who don't feel at ease in traditional classrooms; yet being at home can help them relax [12]. Examples of Synchronous e-learning are online classrooms using audio communication, video communication, instant messaging and more [2], [9].

Advantages and disadvantages of E-learning

Advantages

E-learning provides numerous advantages for both teachers and students, and it is considered one of the greatest ways of education due to its numerous benefits. The following are some of the benefits:

- E-learning is incredibly inexpensive, and the cost of gear and internet has decreased dramatically in recent years. Students can also save money by not driving to and from school [13]–[15].
- Experts can offer advice and exchange ideas with students [13], [15].
- E-learning allows students to take responsibility for their own education [14].
- Students can work from home or from anywhere with access to a computer. This is critical because it offers a conducive working atmosphere for students, some of whom may be hesitant to work in groups or attend school [14], [15].
- Smaller schools, as well as smaller, more rural populations, benefit from e-learning [13].

- E-learning is significantly more adaptive to varied student demands because each student has different strengths and weaknesses [15].

Disadvantages

Despite the benefits of e-learning in education, it does have some drawbacks. The most evident critique of e-Learning is the complete loss of vital human connections between students and professors. Here are a few more examples:

- Students are usually motivated to study harder by the classroom environment and teacher support, and even if they are good students, prizes such as grades can drive them to keep studying. However, some students may lose motivation when participating in E-learning [14], [15].
- When there is no face-to-face communication, some pupils may feel isolated. Finally, the student may drop out owing to loneliness and a lack of guidance and assistance [11], [14].
- Students who do not engage with others do not acquire communication skills, and even if they have good academic knowledge, they may not be able to communicate it to others [15].
- It may be difficult to prevent cheating during online assignments [15].

To summarize, e-learning is one of the fastest growing sectors in education, and it is becoming increasingly popular in higher education institutions due to its numerous benefits. Following that, it will be explained what code testing is and how it works, as it is critical to grasp how it may be used in e-learning.

2.2 Code Testing

Code testing is a collection of tasks that may be prepared ahead of time and carried out in a systematic manner [16], [17]. Code testing is running code in a variety of contexts and determining if software is performing as expected [18]. Testing is done not just to find bugs, but also to see if the product fits client requirements [16]–[18]. We can discover errors using testing in a short period of time and with minimum cost [18]. Testing may often account for up to 40% of a company's software development expenditure [16]. Aside from bug detection, testing allows companies to determine (a) whether their software is ready for integration, (b) whether their software is performant, and (c) improve the overall quality of their software [16].

As mentioned above testing can be expensive to implement therefore it is important to create “good” tests meaning tests that will cover the product sufficiently. Good tests:

- Should have a high probability of detecting an error, this requires the tester to understand the software specifications and attempt to guess where the program could fail [17].
- Tests should be small in size, even though multiple tests can be combined together, it is recommended that this be done only when dealing with complex test scenarios [17], [19].
- Should have none or minimal dependencies to other tests [19].
- Well-documented tests can be used as documentation, or at the very least as a useful supplement to traditional documentation [19].
- Should be designed with maintainability in mind [19].
- Every test should have a distinct function, as repeating the same test is a waste of time and resources [17], [19].

Test driven development

Test-driven development has become increasingly popular in recent years. This strategy involves writing automated tests prior to developing functional code [20], [21]. After writing the automated test cases the code won't even compile since there is no implementation yet. Programmers write the code to pass the test cases [20]. When all of the tests pass, it is assumed that the new code will not create any new bugs, and the code is regarded as suitable for integration [20]. Writing the test cases at the beginning enables continuous testing, which improves code quality and also allows bug detection every time the code is updated [20], [21]. Some of the benefits of test-driven development are:

- Test driven development encourages programmers to try and explain their work using test cases that demonstrate how the code is supposed to behave and by using code itself, as a result it helps in program comprehension [20].
- Thanks to the continual feedback provided to the developer via test driven development, bugs are detected as soon as new code is added [20], [22].
- Software maintenance and bug fixing by patching code is very dangerous because it tends to cause more errors. Some studies even suggest that code patching tends to cause 40 times

more errors than new code. It is possible to determine whether a change disrupts the existing system by continuously running automated tests [20].

- Test driven approach usually results in smaller and easier to understand modules. Usually, the modules produced have less methods per class and overall lower complexity [22].

Types of code testing

Unit and integration testing are two types of code testing. It's best to avoid picking one over the other because they're meant to complement each other.

Unit testing

Unit tests are focused on a "unit" of code. A unit is the smallest software component that can be tested. In software, there is substantial controversy regarding what exactly comprises a unit. Both the class and the method have been recommended as appropriate units in the area of object-oriented programming [17], [21], [23]–[25]. Because systems have become much more sophisticated, that definition appears to be out of date. There has been a lot of effort put towards redefining the term so that it works for both modern and legacy systems. A potential definition for a unit is that it is a component of a complicated system (i.e. a single jsx component in a react app) [25]. A unit test should ignore the rest of the system, meaning it should focus only on a specific unit meaning one or multiple functions [24]. Unit tests could be either positive or negative. Positive tests should check the expected functionality of the unit. Negative tests should make the unit fail and check if the error is handled properly [23]. Unit testing can be challenging at times because (a) identifying testable units in a given software can be difficult, (b) testing units that have or rely on large data structures has been difficult due to the difficulty of setting up the testing environment and (c) creating unit tests without proper documentation can lead to repetition [24].

Integration testing

When all of the different units are merged to make a working program, integration testing is used. Unit testing is done at the statement level, whereas module testing is done at the module level. Integration testing focuses on unit interactions and how effectively they perform together [17], [25]–[27]. Integration testing is the process of ensuring that interfaces between units and modules are consistent and communicate correctly [26]. Integration testing, in the modern era, is defined as the examination of how two or more software components interact (for example, a react app making API calls to a backend) [25]. Because integration tests verify many units in each test, they can be substantially slower than unit tests [25]. During integration testing we are searching for errors that didn't appear during unit testing [27]. Approximately 40% of software errors are detected during integration testing [27]. After a unit is created or modified it should be unit tested before being added to the rest of the software. The results of the integration test will determine whether the module behaves and communicates with the other modules as expected [27].

Why code testing matters

Code testing is an investment that doesn't add immediate value to the final product. It is, nevertheless, an investment that supports other operations that enhance the product's value. The following are some of the reasons why a corporation might wish to invest in developing code tests:

- Customers and software developers gain information on how well software fits mutually agreed-upon requirements [28].
- The knowledge from testing is needed by software developers to ensure that their implementation is done correctly [28].
- Testing aids in the prevention of bugs and flaws [29], [30].
- Aids programmers in truly comprehending the structure of the program [29], [30].
- Tests can be used as documentation, assisting new developers in grasping the current code base's functionality and allowing them to catch up to the rest of the team more quickly [29].
- Developers can improve their programming skills by writing tests [30].

As previously stated, code testing may be used to check and enhance a variety of features of software. These tests can determine whether the software performs as planned or assess the code's quality, such as its structure and design. The following part will cover how to achieve good code structure and design, as well as why these are important.

2.3 Code structure and design

The code architecture of a system specifies how it is broken into components, how they are interrelated, and how they communicate and interact with one another [31]–[33]. Code architecture defines (a) the system's elements and how they communicate [32], [33], (b) composition patterns

[32], and (c) the organization of source code and libraries in the development environment [33]. Software designers are in charge of creating a new system. When a design is new, it is simple, clean, and straightforward. However, when more features are added, the code becomes unkempt, and some software do not even keep their purity after the first release [31]. As new features are introduced, the code grows increasingly, leading to "spaghetti code", meaning it is so complicated that adding new features or updating old ones becomes a significant challenge [31]. Redesigns are frequently proposed, but by the time they are completed, a new redesign is required [31]. Having said that, it's critical to write decent code that's easy to maintain.

Characteristics of good code design

For many years developers and researchers have been looking for ways to design beautiful and more maintainable code. That search has led to the creation of a list with rules of what not to do when designing code. These rules are referred to as bad smells. Bad smells are a result of bad design and poor design implementation [34]. Usually, they are used as indications that the code needs refactoring [35]. By refactoring the code becomes easier to understand and to maintain [35]. Those rules have been evaluated by source code metrics showing that the code is easier to maintain when bad smells are avoided [36]. Some of the bad smells in Object Oriented Programming as reported in [34]–[37] include the following:

- Large methods and classes: methods and classes should not try to do too much. It's tough to understand and manage a class or method with multiple responsibilities.
- Long function parameter lists are more difficult to maintain since they are subject to frequent changes, making them harder to comprehend.
- Duplicate code: To make software easier to extend, duplicate code should be unified. When there is duplicate code, the same modifications must be made multiple times.
- Lazy classes or methods: lazy classes or methods are methods that don't do anything and should either be enhanced in responsibility or deleted altogether.
- Dead code is outdated code that is no longer in use. Dead code should be deleted since it obscures the program structure and makes it difficult to understand.
- Shotgun Surgery: Shotgun Surgery indicates that once one class changes, many other classes should also change. Changes in one class should usually have no effect on other classes.

Why good code design matters

As previously said, badly designed code can have a wide range of effects on software. Developers who maintain and expand software may find it challenging to work with code that is poorly designed. According to [31] “rotten” code can cause: (a) Rigidity: meaning code is very difficult to change even in simple ways, (b) Fragility: meaning software can break in multiple ways when changed, (c) Immobility: meaning the code is not easily reused and finally (d) Viscosity: meaning changes in code cannot preserve the code design rotting the code even more. Many studies have been conducted to determine whether or not bad smells influence code. Those studies interviewed several software engineers after they were confronted with code containing bad smells and code that didn't. These studies revealed the following:

- When confronted with large classes and large methods, programmers complained files were too big and difficult to navigate [38]. They also stated that simpler classes were easier to grasp and more attractive to look at when compared to more complex classes [39].
- When faced with duplicated code they reported they had to consistently make changes [39].
- When faced with feature envy meaning methods that are too much interested in other classes [36] or shotgun surgery they indicated that they made a lot of mistakes since altering a class requires updating a lot of code. They also spent a significant amount of time trying to figure out what needed to be updated [38], [39].

Design Patterns for code quality assurance

The gang of four [40] established design patterns to ensure code quality in 1995, and they've been widely utilized in software engineering since then. Design patterns are standard and reusable solutions to problems in code design that recur frequently [31], [41]–[43]. Design patterns can help make the development process faster by using tested and proven paradigms [41].

In [40] the gang of four divided design patterns into 3 categories depending on their purpose. These categories as described in [40], [44], [45] are:

- Creational design patterns: The use of creational design patterns helps abstract the process of creating new objects. Object creation mechanisms are provided by creational design patterns, which promote the flexibility and reuse of existing code. As systems evolve and become more complicated, creational patterns become more significant because they encourage the reuse of smaller reusable portions of code. Examples of creational design patterns are: Abstract Factory Patterns, Singleton Pattern, Builder Patterns and Factory Method Pattern.

- Structural design patterns: Structural patterns are focused on how smaller units like classes, interfaces and objects are being put together to form larger structures. The utilization of these patterns helps make the structure simpler and easier to understand by identifying the relationships between these smaller units. These patterns also focus on how classes depend on each other. Examples of structural design patterns are Adapter Pattern, Decorator Pattern, Bridge Pattern, Composite Pattern and Facade Pattern.
- Behavioral design patterns: Behavioral patterns are focused on the behavior of objects. These patterns not only indicate patterns of individual units of the code like classes, but also patterns on how they communicate with each other. Chain of Responsibility Pattern, Command Pattern, Observer Pattern, State Pattern, Interpreter Pattern, Iterator Pattern, and other behavioral patterns are examples.

Design patterns, in contrast to code smells, which are utilized as indicators of poor code quality, are employed as proven methods for ensuring code quality [42], [43]. Many researches have been conducted to see if the application of design patterns has an impact on the frequency of code smells. According to a study published in [43]:

- The use of design patterns is often related with the absence of code smells, resulting in more maintainable code.
- Some patterns, such as the Factory method pattern, Adapter-command pattern, and State-strategy pattern, are less likely to be linked to code smells.

Confirming that design patterns and code smells are negatively correlated.

As said, patterns prevent reinventing the wheel, and as a result more patterns have been discovered over time. The Sun Microsystems team defined the J2EE patterns meaning patterns for the J2EE platform. The J2EE patterns include many famous patterns that are widely used today like the Data Access Object Pattern [46]. Another huge attempt in defining more patterns has been made by Martin Fowler in his book Patterns of Enterprise Application Architecture [47]. In this book, Fowler constructed a massive pattern directory with more than 8 categories containing more than 30 patterns in total. Some of the patterns included are the Model View Controller pattern and the

Data Transfer Object which are extremely popular with developers. Many people regard this book to be a go-to resource when it comes to patterns.
Finally, the utilization of patterns has risen in popularity in recent years. As a result, numerous patterns have been identified, with more being discovered every day.

This section discussed good code design and structure. Breaking code into smaller numerous modules with as few dependencies as feasible, as noted several times in this section, is an excellent technique to ensure maintainability. The necessity for maintainability has led to the development of more efficient ways of structuring big codebases. Microservices are a modern approach of structuring server-side applications with maintainability and modularity in mind. More on this new trend and how it can assist developers will be covered in the following section.

2.4 Microservices

Microservices are considered one the newest software lifecycle trend in software architecture[48]. They represent a software and systems design strategy that encourages the usage of small, self-contained services [48]–[51]. These services are responsible for a specific task and communicate with one another via messages [49], [51]. Typically, each service is independent of the others and has its own database and memory space. Microservices were created to address some of the drawbacks of monolithic applications, or applications with a single code base [49]. According to [49], microservices can help with the following issues:

- Due to their size and complexity, large monoliths are challenging to maintain and evolve. Finding bugs necessitates a thorough examination of their vast code base.
- Monoliths are also subject to "dependency hell," which occurs when adding, removing or updating modules or libraries leads to inconsistencies in systems that either do not compile/run or act weirdly.
- Any update to one element of a monolith necessitates a complete reboot of the application. Restarting major projects frequently results in significant downtime.
- Monoliths also limit the technology options available to developers, who must utilize the same language and frameworks as the original program.

The use of microservices helps overcome the above problems because:

- Each service implements a limited set of functionalities, resulting in a compact code base and limiting the amount of code developers must examine in order to find a bug. Furthermore, because microservices are self-contained, a developer can test and analyze their features without regard for the rest of the system [48], [49], [51].
- We have less dependencies on each service as a result of the previous item, which means we can avoid "dependency hell."
- Changing a service does not necessitate a full system reboot; instead, the updated service must be rebooted [49].
- Microservices do not impose any additional lock-in, and developers are free to choose the best technologies for each service, such as language and framework [49].

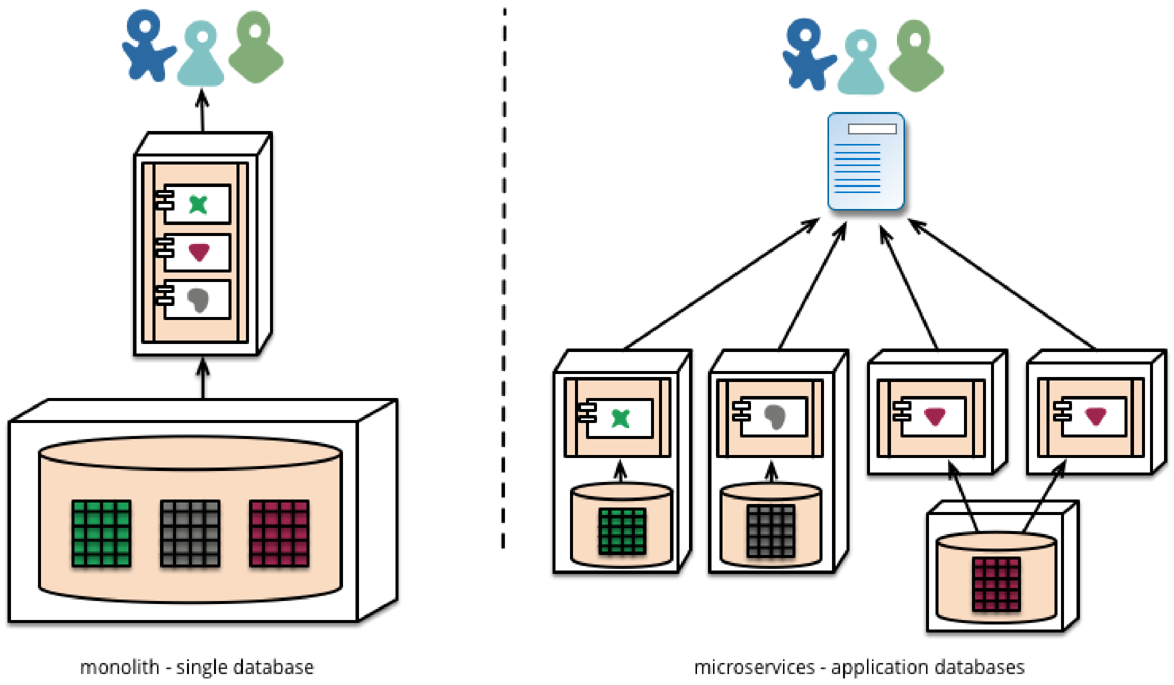

monolith - single database
microservices - application databases

Figure 1: Microservices architecture [52]

Challenges of microservices architecture

Because the utilization of microservices is still relatively new, there are several outstanding concerns and optimization opportunities in this field [53]. A lot of research has been done to identify the challenges of creating applications using microservices with the intent to develop ways to overcome them. Reading through [53]–[56] some of the identified challenges are:

- Debugging and maintaining several modules, especially those that rely on other modules, can be difficult.
- Managing shared code across numerous services is challenging because each change has the potential to break multiple services unintentionally.
- Decomposing monolithic codebases into smaller services is a challenging operation that can result in several issues if not done appropriately.
- Because most developers are accustomed to a monolithic mindset, finding skilled developers can be difficult.
- It takes a long time and a lot of money to change the mindset of existing engineers who are used to creating monolithic apps.
- Collaboration amongst teams, each of which is working on a distinct service, is a difficult task that necessitates a solid foundation of team communication.
- Preventing security threats is much more difficult since more services means more exposed surface for attacks.

As microservices become more widely utilized, there will be more efficient approaches to address the issues listed above.

Advantages and disadvantages of using microservices

Microservices can provide several benefits to the development process, but they also come with a number of drawbacks due to the complexity of the microservices design.

Some of the benefits of using microservices are:

- Increased Maintainability: Services are smaller, making maintenance easier [57]–[59].
- Microservices are often delivered independently from the rest of the application, whereas monolith teams must coordinate in order to deploy together [57], [58].

- Simple Scalability: Microservices are easier to scale than monoliths because of their nature [58].
- Self-Healing: Older safe and tested versions can replace newer failed services [57], and due to its stateless qualities, a service can be easily restarted when it fails [59].
- The introduction of a library or component including shared features increases code reusability by lowering maintenance effort because the shared features will only be modified once, and any associated microservices will be updated as well [57].
- Flexibility: Each team is free to choose their own technology based on their requirements. [57].
- Understandability of software: Developers can better understand what each service does with smaller codebases [59].

Some of the disadvantages of microservices are:

- Testing Complexity: The complexity of testing increases when more services and communications between them are added [57].
- The shared code registry, if not built properly, could be the system's primary bottleneck [59].
- Senior developers are required: Architectural considerations are difficult, and beginning the creation of a microservices-based system necessitates the use of highly competent developers [58].
- Requirement for automation: To manage the entire system and automate the entire process, a full DevOps stack is required. The process of developing a system would be substantially slower when using microservices compared to monolithic systems if DevOps was not adopted [57].

3 Background work

This chapter surveys previous work in online platforms for automatic code grading. Coding challenge platforms are a subset of the larger subject of education and programming training, and they are considered as an important resource in computer science. Such platforms offer diverse challenges, with a particular emphasis on the algorithms and data structures that must be maintained while students constructing a solution to a problem that must be reviewed in order to test programming skill.

3.1 Topcoder.com

TopCoder is one of the first online platforms for competitive programming. TopCoder.com was first introduced in 2001. It currently has over 1 million registered developers and has created software for Fortune 100 corporations, worldwide enterprises, and government entities. Topcoder compensates community members for their contributions to projects and sells community services to corporations, small businesses, and nonprofits. A customer can use topcoder to (a) solve problems by organizing a competition in which participants compete for money, (b) employ freelance developers from all fields of computer science, and (c) have projects finished by both freelancers and developers competing against one another. Developers are compensated depending on the number of competitions they win, therefore payment is not guaranteed. However, they are encouraged to join in order to enhance their skills rather than to get compensated. Topcoder provides UI/UX design, QA and testing, data science, web development, and other services and solutions to businesses [8], [60].

3.2 Coderbyte.com

Coderbyte is an online system that provides a collection of programming challenges and courses with the goal of assisting users in preparing for forthcoming job interviews as well as increasing their programming skills. The difficulty of the code challenges varies, and Coderbyte has its own IDE built into the web application. As a result, the user can solve the problem in its online editor and observe if the answer is executed successfully in real time. Coderbyte offers over 200 coding tasks that users may complete online in one of ten programming languages. The challenges range from simple (finding the longest word in a string) to difficult (print the maximum cardinality matching of a graph). They also have a number of algorithm tutorials, introductory videos, and interview preparation courses available. Companies can also use Coderbyte's services, such as conducting remote, real-time interviews and code-pairing with candidates in a live programming environment. They also provide companies the option to assign tasks to candidates [7], [61].

3.3 Codewars.com

CodeWars is an online resource that gives users a variety of activities to complete that require them to use their coding skills. Many popular programming languages are supported by this platform. Most tasks were generated by the community, meaning its users. Users are rated based on how many challenges they solve. After solving a task users are provided with a preferable solution based on evaluation. After the assignment has been completed, the user can review the most preferred and highly rated solutions. Companies can use Codewars' technical skills tests, and educators can use Real-Time Pair-Programming and Programming Tests to teach and test their students objectively [6], [62], [63].

3.4 Codingame.com

CodinGame is a game-based learning tool for learning programming languages. Its methodology is focused on puzzles that have multiple animations with characters and fun sequences that students can watch when they solve the challenge. The game includes a problem description, test cases, and an editor that allows visitors to develop code in one of 20+ programming languages. Despite the fact that this website is not like the others mentioned above in terms of competitive programming, it is nonetheless popular among programmers who enjoy solving challenges and competing. CodinGame contests are online programming competitions in which software developers and programmers from all over the world participate for fun or to connect with organizations that are hiring. The goal of CodinGame is to create computer programs that handle complex tasks in a short amount of time. The platform's goal is to help participants and employers connect for selection, as well as to give a pleasant approach to learn, practice, and improve programming abilities. [5], [61], [64], [65].

3.5 Hackerrank.com

Algorithms, Mathematics, SQL, Functional Programming, AI, and other fields are among the challenges offered by HackerRank. Users can use their IDE to solve all of the problems immediately online. When a programmer submits a solution to a programming challenge, the accuracy of their output is evaluated. Programmers are then ranked globally on the HackerRank scoreboard and awarded badges for their achievements, in order to encourage people to compete. In addition to individual coding tasks, HackerRank also runs contests (dubbed "CodeSprints") in which users compete on the same programming problems over a fixed period of time and are ranked at the end. Companies wishing to hire experienced developers can also use Hackerrank [4], [66], [67].

3.6 Leetcode.com

LeetCode is a well-known portal that offers 2300 exercises to assist users prepare for technical job interviews. Users can complete the problems in one of 14 programming languages immediately online. Users, on the other hand, are not able to see other people's solutions, but they are given data of their own, such as how fast their code ran in comparison to other people's code. LeetCode also features a Mock Interview part dedicated to job interview preparation, as well as their own coding contests and a portion of articles to help users better comprehend specific difficulties [3], [68].

Bellow is a comparison table of all the platforms that are mentioned above. This table compares the platforms based on the languages they support and the basic features they offer.

Features / Tool	Helps companies recruit developers	Supports competitions	Anyone can create challenges	Supported programming languages	see other people's solutions
TopCoder	Yes	Yes	Companies	10+	Yes
Coderbyte	Yes	No	Companies	10+	No
CodeWars	Yes	Yes	Yes	10+	Yes
CodinGame	Yes	Sometimes	Yes	10+	No
Hackerrank	Yes	Yes	Yes	10+	Yes
LeetCode	Yes	Yes	No	10+	No

Table 1: Comparison of similar platforms

4 Software Design

This section focuses on the problem that this thesis is attempting to tackle as well as the recommended solution. The problem will also be examined and broken down into use scenarios in this part. Finally, based on these scenarios, the complete platform design, including architecture, major components, and features, will be presented.

4.1 Problem Description

With the rapid rise in popularity of computer science and the high demand for software engineers the number of individuals interested in the field has increased. This has had an impact on educational institutions such as universities, which have been straining to keep up with demand [69]. Grading several complex programming assignments is one of the issues that professors face because it is time-consuming, expensive, and takes time away from other high-value contributions that professors could provide [70]. Another major issue is that, despite the rising need for developers, many people lack access to educational institutions, primarily due to cost, resulting in an increase in demand for alternatives. However, despite the demand there is a lack of resources to gain or improve programming skills. Online coding learning platforms are a well-known and cost-effective option [3]–[8]. Coding challenge platforms are a subset of the broader field of training and education programming support, and they play a critical role in computer science. Such platforms provide a wide range of challenges, with the main focus on the algorithms and data structures that must be managed while composing a solution (algorithm) to a problem that has been submitted by the student and must be evaluated in order to evaluate programming knowledge [66]. Educational institutions can also use such systems to automate the time-consuming process of code grading [70]. According to [71] continuous assessment can make studying more effective for students. Even experienced programmers can use such platforms to challenge themselves and develop their skills by completing difficult code problems.

With this in mind, a platform is proposed for managing and grading programming tasks. It allows professors to create programming assignments for students to complete and grade them based on input output tests, structural checks, and even design pattern checks. It can also be utilized by people who want to learn or improve their programming skills by participating in challenges created and maintained by the community.

4.2 Software Analysis

The needs from the end user's perspective will be defined in this section. The definition of requirements is a critical stage in determining what needs to be done to meet the demands of users and stakeholders. In the following sections, these requirements will be utilized as a guideline for designing the system.

4.2.1 Platform’s scenarios of use

User stories are brief, informal summaries of software features written from the end user's point of view. User stories are mostly used in agile software development to express the value that a product feature can provide and to gain a deeper understanding of why users want a certain feature. These user stories are usually written on index cards or sticky notes by the development and product management teams and used in discussions during the development process. These discussions allow the team to concentrate on the consumer at the center of the conversation, resulting in a stronger product and user experience. User stories are categorized based on the various roles that each user can have. A user can have one of four roles in this app: guest, student, instructor, or administrator. Because each role necessitates the implementation of different functionality, each role has its own set of user stories. Each user story has one or more acceptance criteria, which are a collection of specified requirements that must be completed for a user story to be considered complete.

<u>Role guest user stories</u>

Guests are users who have not been authenticated. Below is a list of the user stories for guests as well as their acceptance criteria.

- As an unauthenticated user I want to register to the platform using my personal info so the system can authenticate me and remember me.
    - Given that I am an unauthenticated user, and I am on the register page, when I enter valid credentials then the system will create my account.

    - Given that I am an unauthenticated user, and I am on the register page, when I enter invalid credentials then the system will show me an error.
- As an unauthenticated user that has an account on the platform I want to log in using my email and password so that the system can authenticate me.
    - Given that I am an unauthenticated user, and I am on the login page, when I enter correct email and password then the system will sign me in.
    - Given that I am an unauthenticated user, and I am on the login page, when I enter invalid credentials then the system will show me an error.

<u>Role teacher user stories</u>

Teachers are authenticated users. Teachers and students have the same role since every authenticated user can create or participate in an exam. However, to make the user stories clearer, teachers and students will have different stories. Below is the list of teacher user stories and their acceptance criteria.

- As a teacher I want to create private challenges so I can test my students
    - Given that I am an authenticated user on the create exam page when I enter valid exam details, the system should create a private challenge.
    - Given that I am an authenticated user on the create exam page when I enter invalid exam details, the system should show an error.
- As a teacher I want to edit my private challenges so I can fix mistakes or update them regularly
    - Given that I am an authenticated user on the edit exam page that I own when I enter valid exam details, the system should update the private challenge.
    - Given that I am an authenticated user on the edit exam page that I own when I enter invalid exam details, the system should show an error.
- As a teacher I want to delete my private challenges so that I can get rid of outdated classes or challenges accidentally created.
    - Given that I am an authenticated user on my profile page when I press the delete button on a challenge, I own the system should delete it.

- As a teacher I want to view statistics about my challenges, so I know how my students performed
    - Given that I am an authenticated user when I click on a challenge, I own then the system should give me statistics on the overall performance of my students.
- As a teacher I want to download code submitted by students so I can manually review it and keep it for record
    - Given that I am an authenticated teacher on the challenge statistics page when I click on a user submission the system should download the code.
- As a teacher I want to submit requests to admin to create public challenges so I can contribute to the community
    - Given that I am an authenticated user on the create exam page when I enter valid exam details and select the public option, the system should create a create exam request for an admin to review.
    - Given that I am an authenticated user on the create exam page when I enter invalid exam details and select the public option, the system should show an error.
- As a teacher I want to submit requests to admin to edit my public challenges so I can fix mistakes or update the exam
    - Given that I am an authenticated user on the edit exam page that I own when I enter valid exam details, the system should create a create exam request for an admin to review.
    - Given that I am an authenticated user on the edit exam page that I own when I enter invalid exam details, the system should show an error.
- As a teacher I want to submit requests to admin to delete my public challenges so I can remove outdated classes
    - Given that I am an authenticated user on my profile page when I click delete challenge, the system should create a delete exam request for an admin to review.
- As a teacher I want to override old edit requests so I can fix mistakes or add more features
    - Given that I am an authenticated user on my profile exam page, when I click on edit challenge for a challenge that I already have submitted an edit request, the system should ask me whether I want to load the previous request as boilerplate.

    - Given that I am an authenticated user on the edit page for a challenge that I own that has a pending request, when I click on submit, the system should replace my previous pending edit request.
- As a teacher I want to cancel pending requests so I can cancel requests that add no value or accidentally submitted requests
    - Given that I am an authenticated user on my profile page at the pending requests section, when I click on cancel pending request, the system should cancel my request and replace it with my previous pending request.

<u>Role student user stories</u>

As mentioned above authenticated users are both students and teachers meaning every user can create and participate in exams. Below are the user stories for students.

- As a student I want to browse all the public challenges so I can select the one that fits me best
    - Given that I am an authenticated student, when I go to the practice page then the system should show me a list of all the available public challenges.
- As a student I want to join a public challenge so I can practice my programming
    - Given that I am an authenticated student at the practice page, when I click on a challenge, the system should redirect me to the exam page.
- As a student I want to run my code so I can receive feedback
    - Given that I am an authenticated user, and I am on the exam page, when I click the run button the system should give me feedback about my solution so far.
- As a student I want to submit my code so I can get it graded
    - Given that I am an authenticated user, and I am on the exam page, when I click the submit button the system should submit my solution and give me my results.
- As a student I want to browse my challenge history so I can track my progress
    - Given that I am an authenticated user when I visit my profile page the system should show me a list of my exam history.

Role admin user stories

Admins are users with special privileges that moderate the application and its content. Admins are also responsible for approving or declining incoming requests from teachers for the public challenges. The list of admin user stories is:

- As an admin I want to view all challenges so I can review and monitor them
    - Given that I am an admin on the admin panel, when I click on the challenges tab the system should load a list of all the challenges.
- As an admin I want to edit any challenge so I can make sure they are correct and proper according to the website rules
    - Given that I am an admin on the challenges tab, when I click on edit a challenge and submit my changes, the system should update the challenge.
- As an admin I want to delete any challenge so I can get rid of challenges violating the website rules
    - Given that I am an admin on the challenges tab, when I click on delete a challenge, the system should delete the challenge.
- As an admin I want to view all requests so I can monitor user requests
    - Given that I am an admin on the admin panel, when I click on the requests tab the system should load a list of all the requests.
- As an admin I want to approve good requests so public challenges are always up to date and as good as possible
    - Given that I am an admin reviewing a pending request, when I click on approve, the system should approve the pending request.
- As an admin I want to decline bad requests so I can prevent useless changes and changes violating the website rules
    - Given that I am an admin reviewing a pending request, when I click decline, the system should decline the pending request.
- As an admin I want to be able to view statistics about the website, so I know how the app performs
    - Given that I am an admin, when I visit the admin panel, the system should provide some statistics.

- As an admin I want to be able to view statistics from all the challenges so I can see how students perform
    - Given that I am an admin, when I visit a challenge details page, the system should provide info on the system.

4.3 Eurytus Design

The general process of developing the platform will be discussed in this chapter. The system was created to address the issues and requirements outlined in the preceding chapters. The system's functions and aims will be detailed, as well as how the system was broken down into several services to achieve them. There will be diagrams of the overall platform as well as diagrams for each component.

4.3.1.1 Architecture

In Figure 2 the main architecture of the toolset is displayed. The system is composed of 5 backend services, an event bus, a load balancer and a react app. Each microservice consists of a backend app and a database. Because all microservices are independent of one another and are deployed individually, if one fails, the others are unaffected. Because the microservices are self-contained, they do not communicate directly with one another. Instead, an event bus is developed and used to handle event-based communication. A load balancer is utilized to distribute http requests among all microservices in order to access each one. Finally, a react app is created to serve as the user interface for the application that communicates with the load balancer in order to access each service.

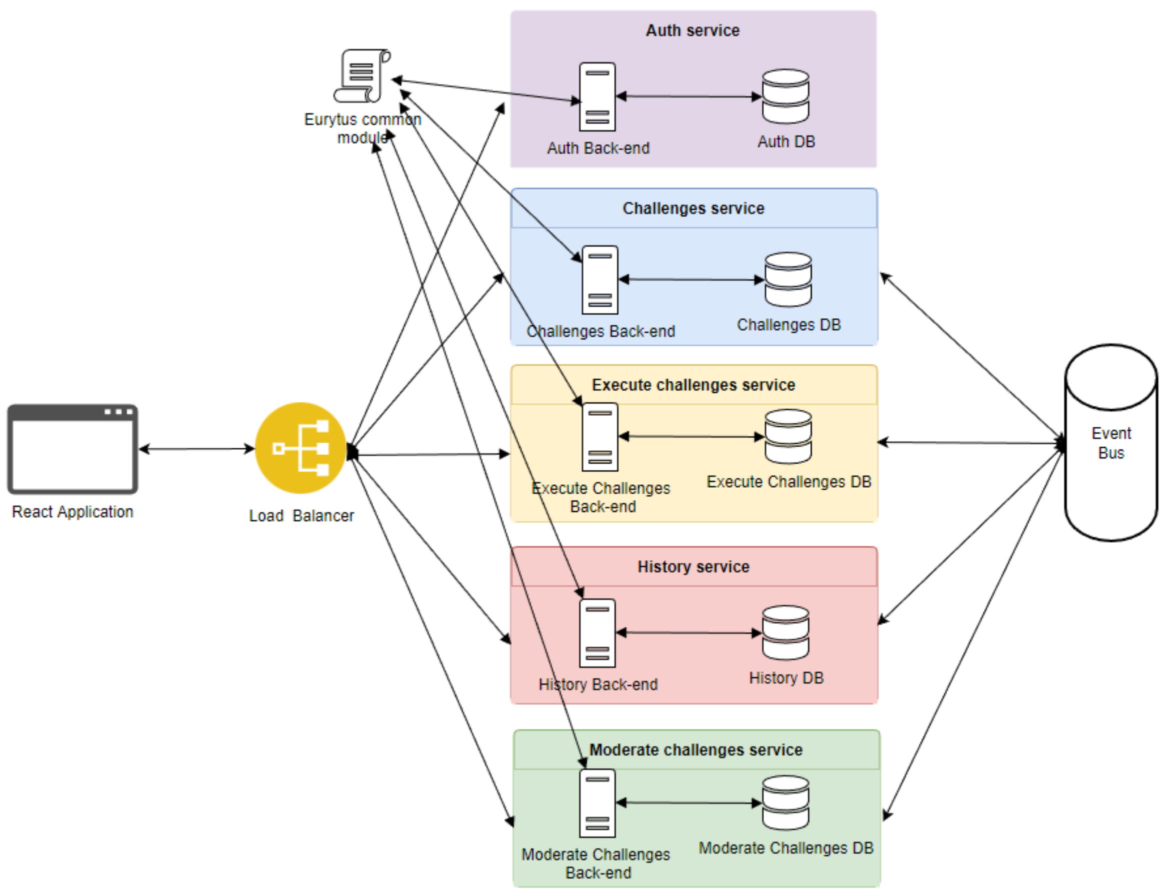


Figure 2: High level Architecture Diagram

4.3.1.2 Component diagram

The components used to deliver the required functionality are shown in this section. Component diagrams also show the connections between the various components and how putting them together composes the application. The application is divided into several components, each of which is dependent on other components and in turn is dependent on them. The component diagram contains all the components, as well as their dependencies and relationships. Because the app has several components, presenting the entire component diagram as a single image would be too large and cluttered, making it difficult to understand. Rather, a series of diagrams will be utilized to depict the complicated components and their interdependencies. Finally, the final diagram will be provided, indicating the operation of the entire system as well as where the complex components indicated above are used.

4.3.1.2.1 Eurytus

Finally, in figure 3 The overall component diagram of the app is presented. The components described in detail above will be presented without their dependencies to make the diagram easier to understand. The web app component mentioned above depends on the load balancer which is a subsystem that distributes the http requests to the microservices. As seen in the figure below the load balancer depends on all 5 microservices. The 5 microservices of the system are: (a) Challenges service that handles the creating, updating and deleting challenges, (b) Auth service that handles the authentication of the users, (c) Moderate challenges service that is used to moderate all public challenges by the admins, (d) History service that is used to save the grades of all submitted assignments and create statistics for each challenge and finally (e) Execute challenges service that is described above. All these services depend on the Eurytus common module that has previously been explained in greater detail, again as mentioned above the reason why all these services depend on this module is to avoid repeated code among all services. Finally, the final component of the application is the Event bus service that is a dependency for all services except the authentication service and is used by all of them to communicate with each other using an event-based system.

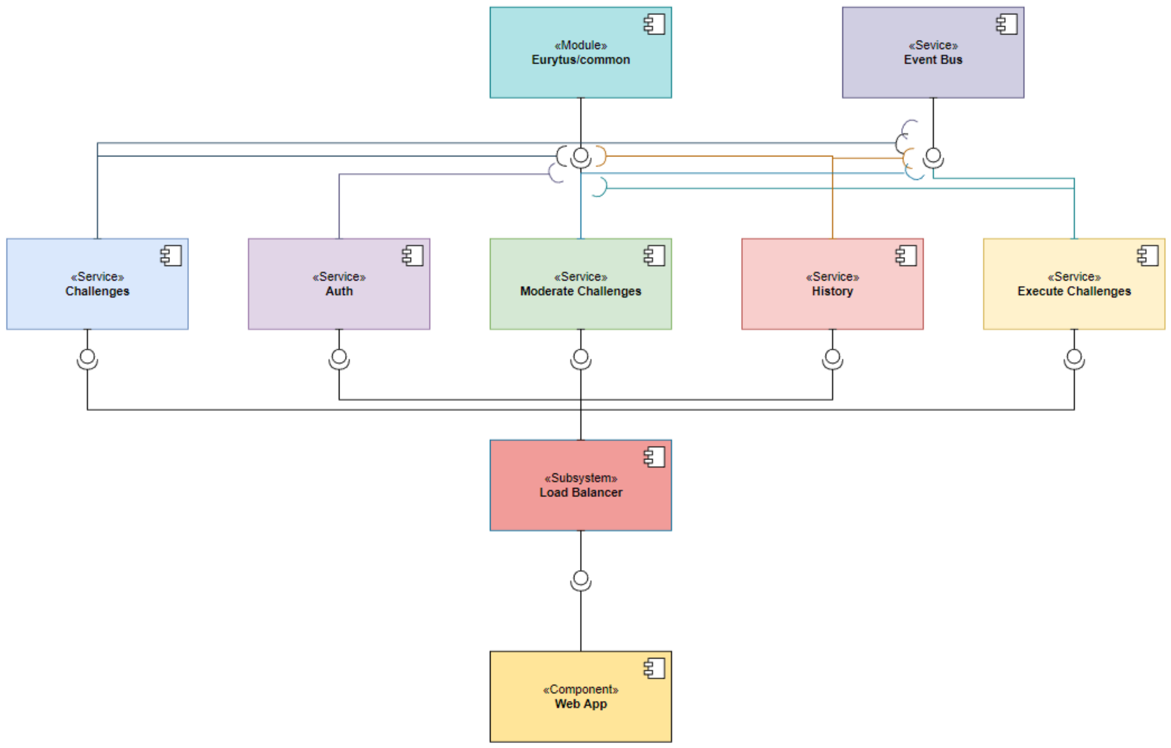


Figure 3: Eurytus components

4.3.1.2.2 Eurytus common node module

In figure 4 the component diagram of the Eurytus common node module is displayed. The Eurytus common module is a node module that contains all the common functionalities between services. The reason why this node module was deemed necessary in the first place is because in micro-services applications there is a lot of repeated code among all services (like functions to get the current user or error handling functions). By using a node module, the code is written once and updated once. To use the shared functions, the node module is published to a library registry like npm, and all services must import the most recent version of the package. The Eurytus common module includes authentication methods, error handling methods and event handling methods. All these functions are regularly utilized across all services, and they would have to be written and changed several times if not for this module.

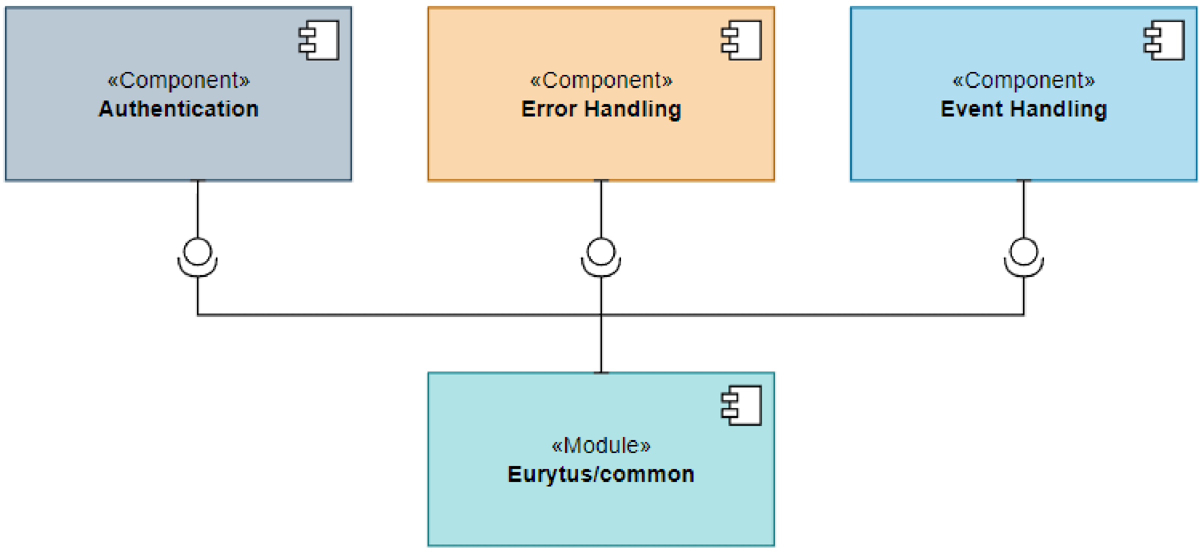


Figure 4: Eurytus common components

4.3.1.2.3 Execute challenges service

Next, the components that compose the execute challenges service are presented using a component diagram in figure 5. The reason why this service is presented separately from the other services is because this service handles a lot of the functionalities offered by the app and as a result contains more complex components that require explanation. This microservice receives code from the users and handles the Code execution, Design pattern detection, Code structure checking, and the assignment grading based on the results of these tests and the requirements specified by the teacher. The service consists of a module called Eurytus/compile-run which is a node module developed specifically for this project and is based on an already existing open-source package called compile-run. This module handles the code execution of the code submitted by the users and returns the compiler's output. The service also includes a component for checking code structure, which is exclusively utilized for java assignments. This component is used to see if the structure of the code that has been provided matches the structure that the teacher has specified. Finally, the Design pattern detection component is required by this microservice. This component is only used for java assignments and only if the teacher has specified design patterns that must be implemented. This component takes the code structure of the submitted code and checks if it matches the structure of the required design patterns.

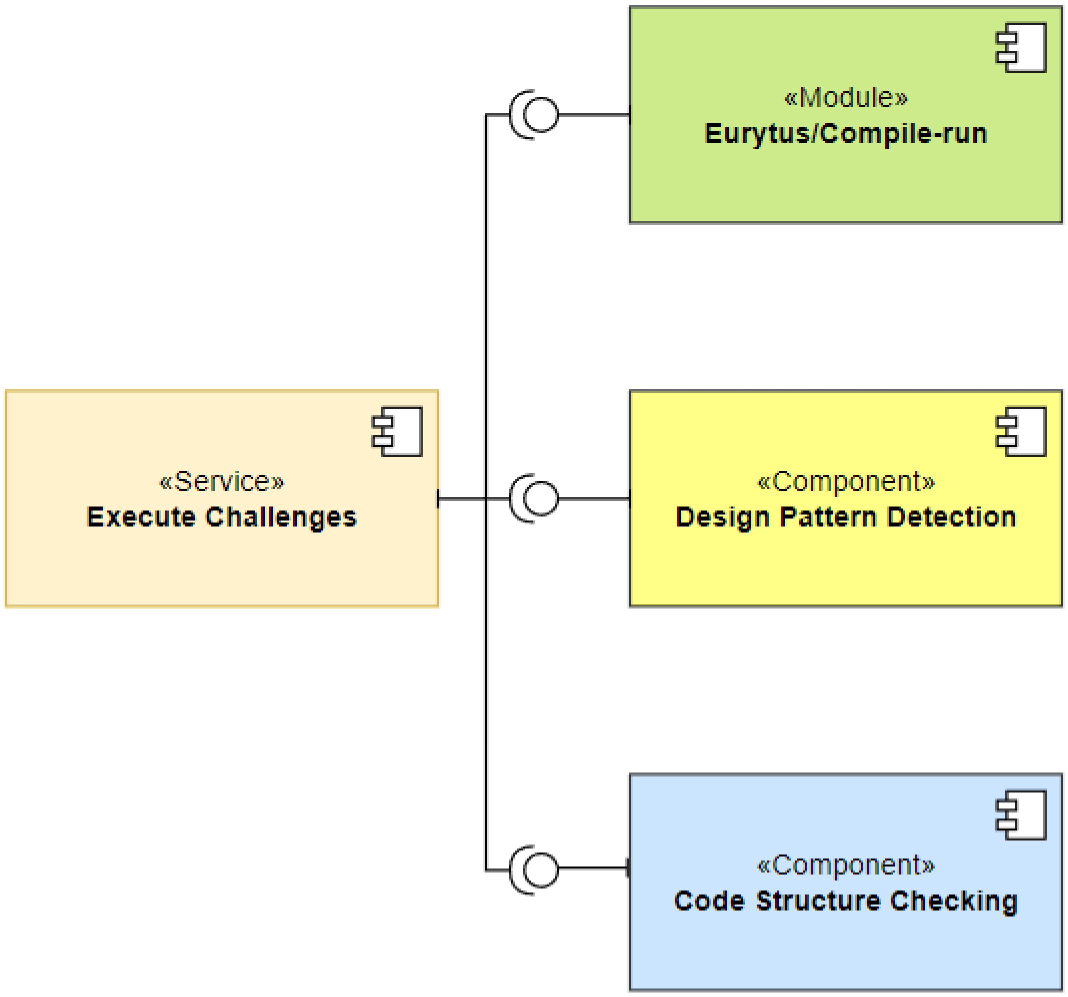


Figure 5: Execute challenges service components

4.3.1.2.4 Eurytus UI

The component diagram of the Web app is shown next in figure 6. The user interface of the app is provided by the web app component, which is where users interact with the services when writing code, submitting code, exploring challenges, and so on. This component is dependent on an IDE module, which users use to write code, and a Class builder component, which was built as a more interactive alternative to specifying the needed code structure of an exam. Both components will be presented as implemented in the app in Chapter 6.

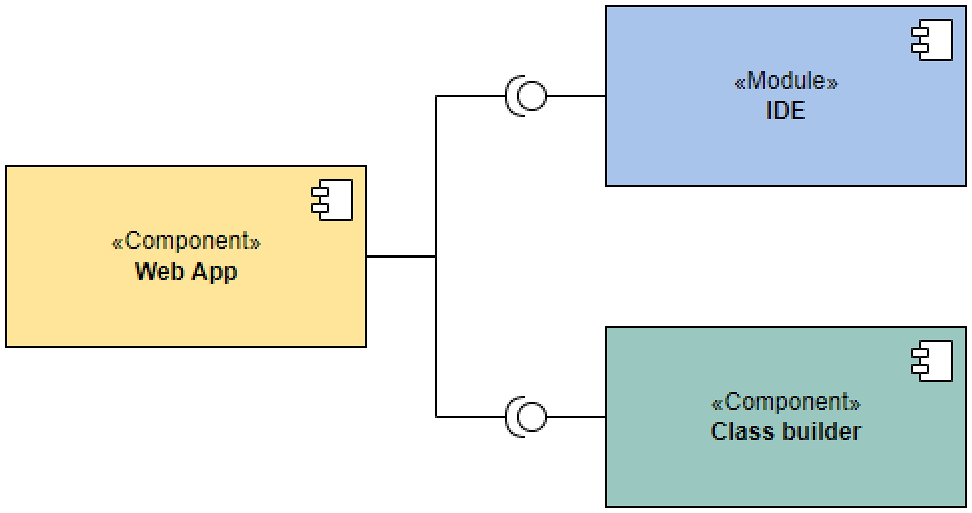


Figure 6: Eurytus UI components

4.3.2 Eurytus' s functionalities, procedures, and events

Flow diagrams are visual representations of connected information such as events, process steps, functions, and so on. The flow diagrams that arise depending on the requirements stated in the user stories above will be presented in this section. Each diagram will cover a functionality rather than a user narrative, hence there will be several user stories in each diagram. In the next pages the most important functionalities of the Eurytus platform will be elaborated.

4.3.2.1 Authentication flow diagram

The flow diagram of the authentication procedure is shown in figure 7 Users that want to authenticate will go to the authentication page. Then, if they've previously signed up, they'll need to login using their credentials. They will need to register a new account by providing their information if they are new users. The user will be regarded as authenticated and logged in after either step is completed.

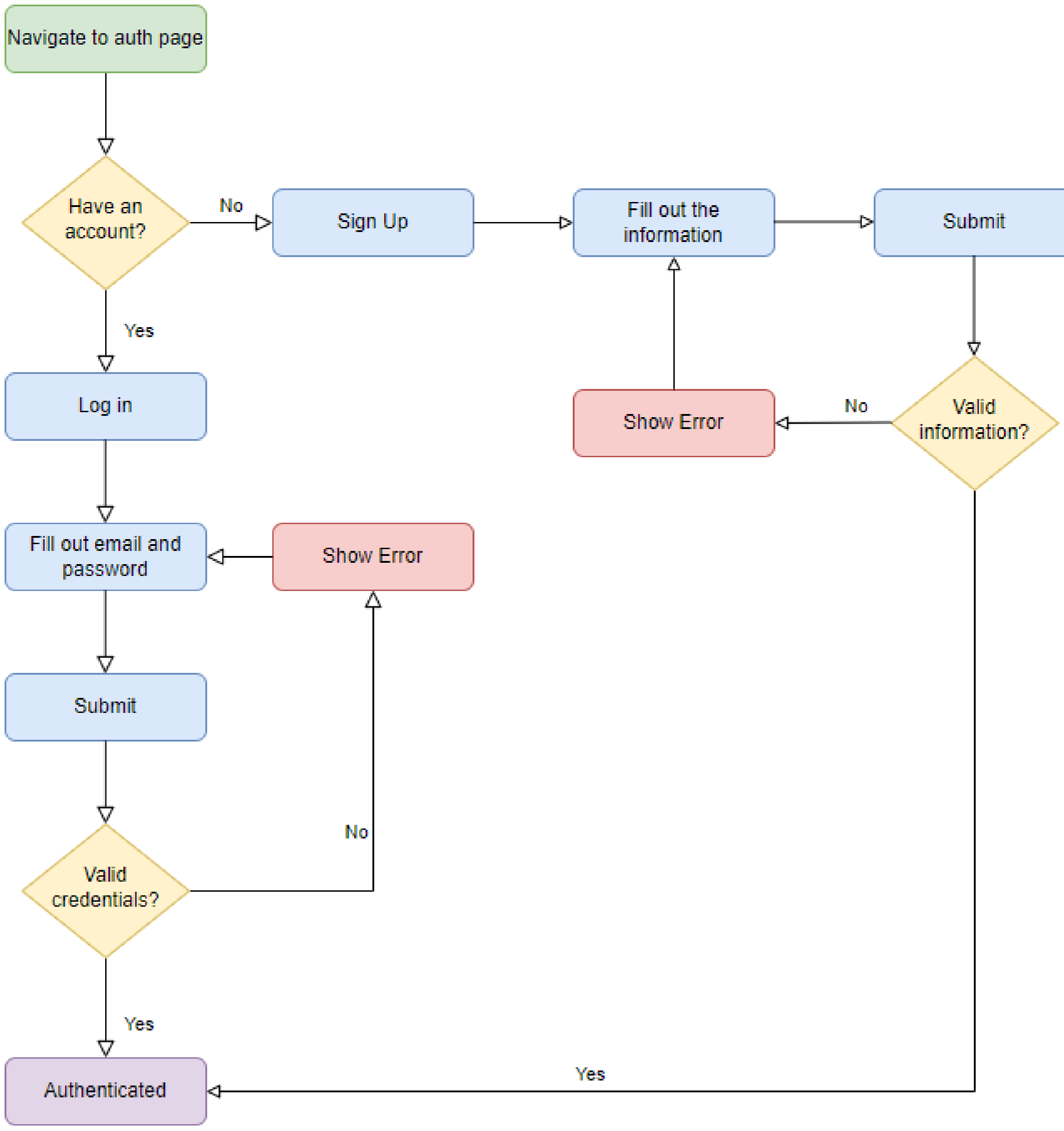


Figure 7: Authentication flow diagram

4.3.2.2 Create challenge flow diagram

Following that, as shown in figure 8, the procedure of establishing a new challenge will be outlined. Teachers must first fill out proper exam information such as the name, description, code template, tests, and so on. After that, the user must choose whether the challenge is public or private. If the challenge is private, the challenge is created, if the challenge is public, a request to create a challenge is created, which must be approved by an admin.

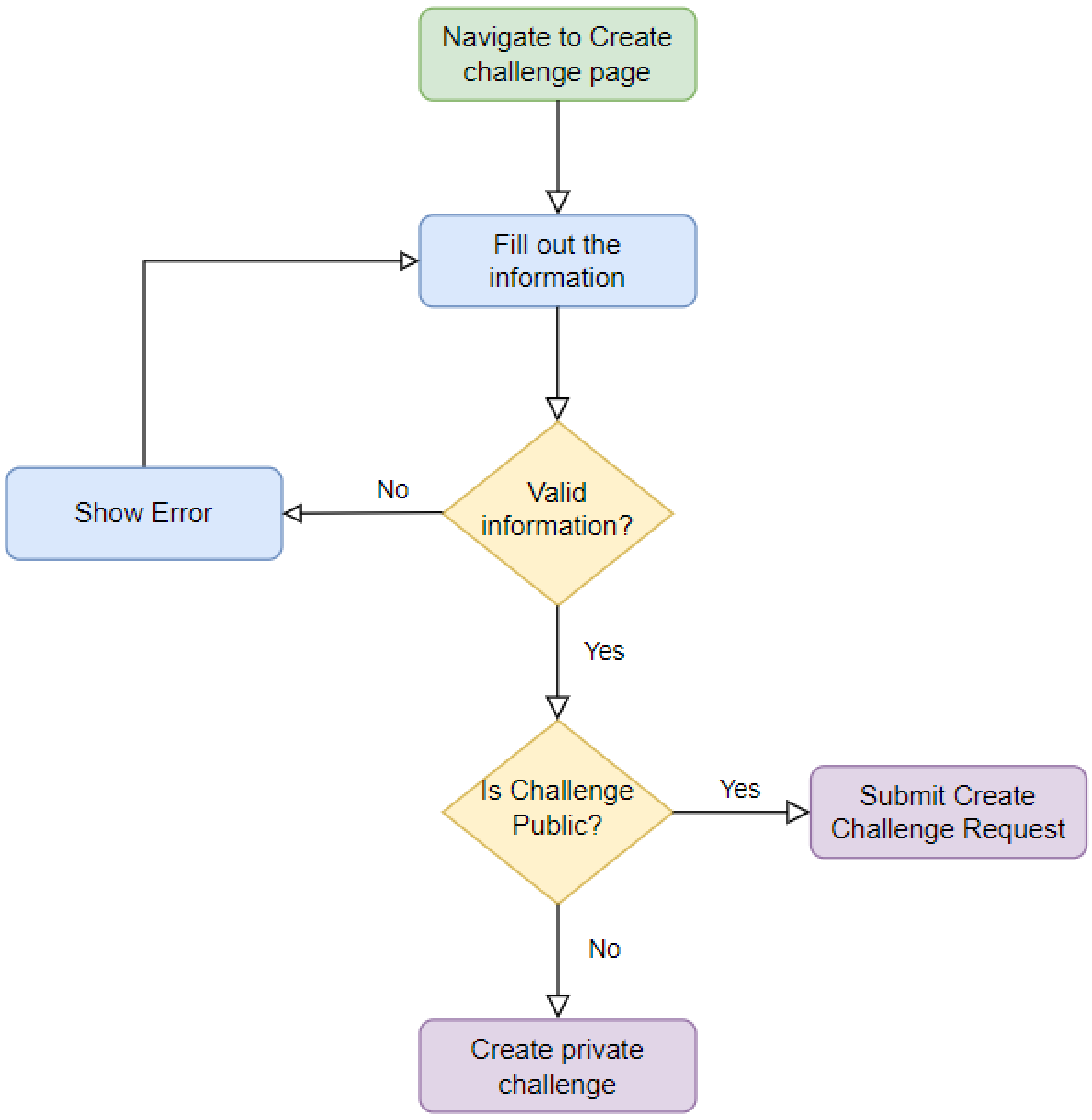


Figure 8: Challenge creation flow diagram

4.3.2.3 Edit challenge flow diagram

Next, the figure below shows the flow diagram of editing a challenge. The behavior of the program when a user modifies a challenge varies based on whether the challenge is public or private. If the challenge is private and the data provided is valid, then the challenge is updated. If the challenge is open to the public, all changes must be approved by an administrator. This means that the teacher might make additional adjustments while waiting for some revisions to be approved. When a user tries to edit a public challenge, the system looks for pending requests because of this issue. If a pending edit request exists, the user can either load and modify the prior pending request or update the current version of the challenge. If the pending request is a delete challenge request, the user will receive an error message requesting that the pending request be canceled first. Finally, when a user modifies a challenge, a request to edit the challenge is made to the admin for approval.

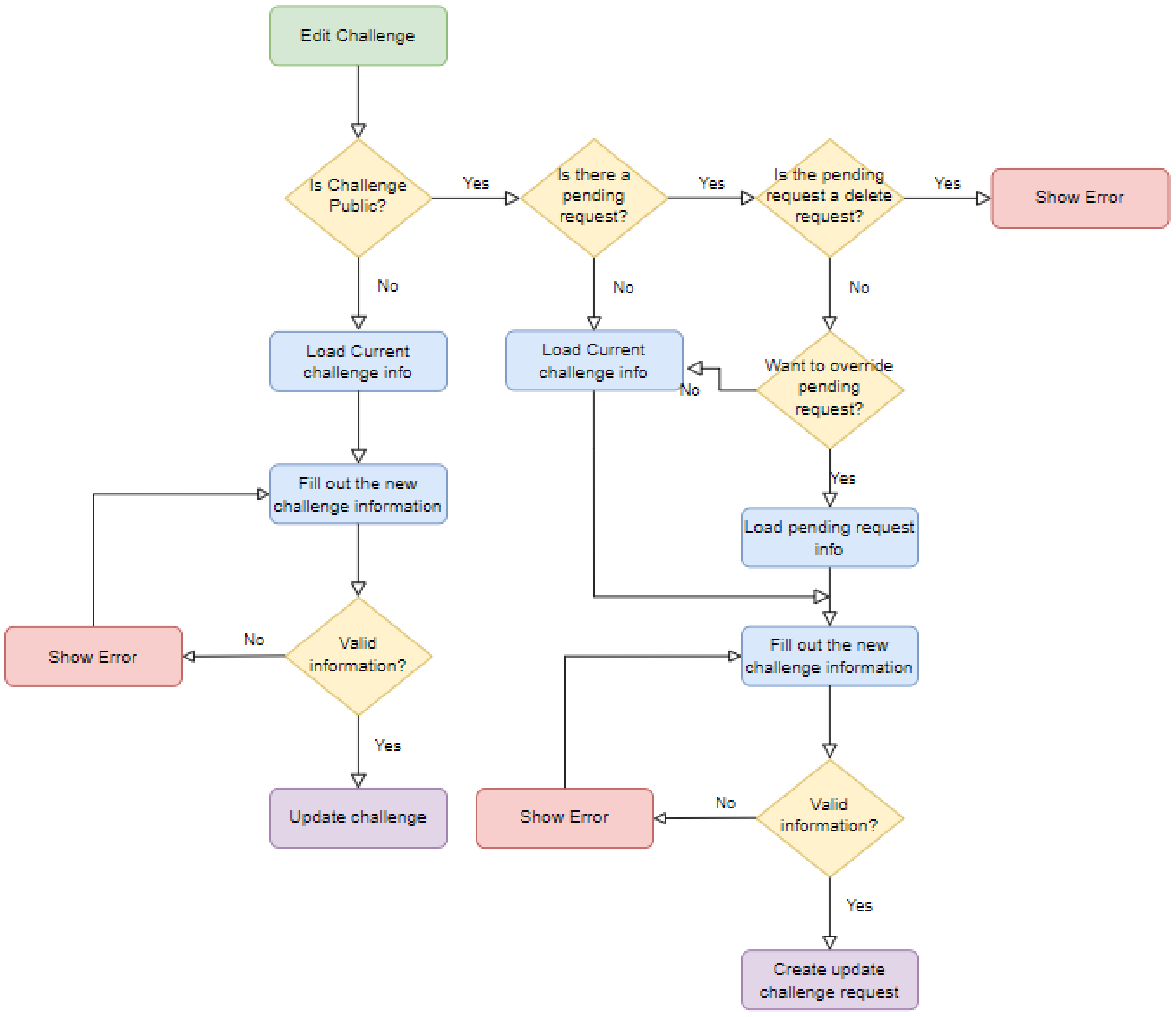

Edit Challenge
Is Challenge Public?
Yes
Is there a pending request?
Yes
Is the pending request a delete request?
Yes
Show Error
No
No
No
Load Current challenge info
Load Current challenge info
Want to override pending request?
No
Fill out the new challenge information
Yes
Load pending request info
Show Error
No
Valid information?
Fill out the new challenge information
Yes
Update challenge
Show Error
No
Valid information?
Yes
Create update challenge request

Figure 9: Edit challenge flow diagram

4.3.2.4 Delete challenge flow diagram

The delete challenge flow diagram is depicted in figure 10 When a user wants to delete a challenge, the procedure varies based on whether or not the challenge is public. The user can erase the challenge if it is private. If the challenge is public, however, a request to delete it is created and sent to the administrator for approval.

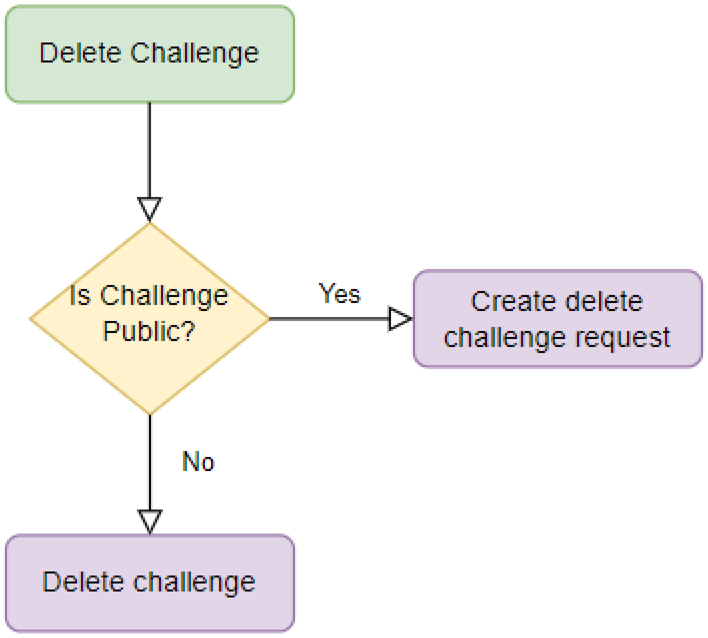


Figure 10: Delete challenge flow diagram

4.3.2.5 Solve challenge flow diagram

The flow diagram for completing a challenge is shown below. When users join a challenge, they must complete the code according to the challenge's requirements in order to complete it. Users can run their code numerous times to see how far they've progressed and make any necessary changes. Finally, users can submit their solutions in order to save their grades.

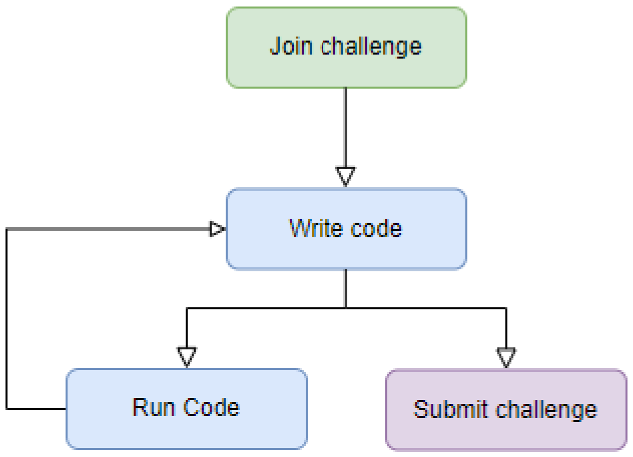


Figure 11: Solve challenge flow diagram

4.3.2.6 Request moderation flow diagram

As previously stated, users must obtain admin clearance before they can create, edit, or delete public challenges. As a result, an administrative approval mechanism must be implemented. In the diagram? There is a flow diagram of the request moderation system presented. Admins receive pending requests, which they must analyze to determine whether they are worthy of approval. Finally, they can accept or reject the pending request.

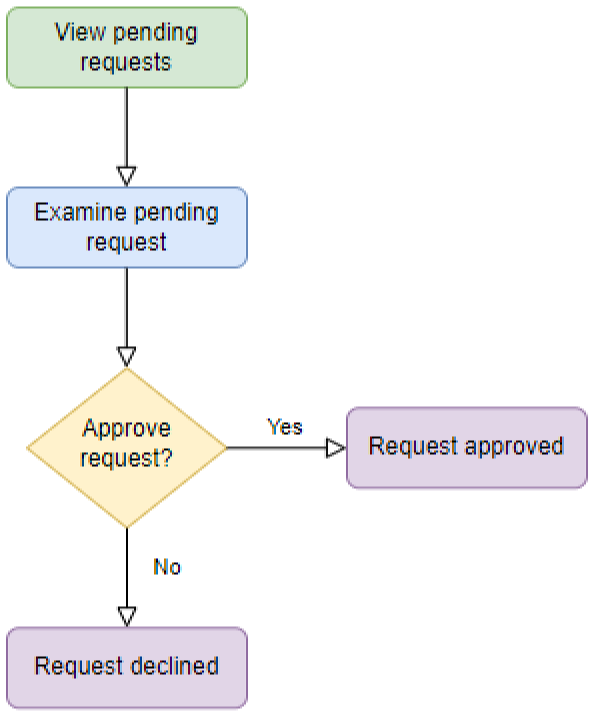


Figure 12: Request moderation flow diagram

5 Software Implementation

This chapter focuses on the Eurytus platform's implementation. This section begins by providing an overview of the technologies utilized to implement the Eurytus platform and why they were chosen. Finally, the implementation of the core capabilities will be presented, along with helpful code snippets and graphs.

5.1 Technologies used

A brief review of all the technologies used to implement the application will be presented in this section. Eurytus is a platform that is built for long-term performance and operability, including continuous deployment, easier maintenance, and high scalability. To fulfill these objectives, a range of cutting-edge open-source technologies were utilized, which allow for easier platform expansion including replacement of current technologies with newer or better technologies in the future.

5.1.1 Docker

Docker is an open-source platform that not only helps making the development process easier but the distribution process too. Docker-based apps are bundled and executed in virtual environments called containers that include all necessary dependencies. These containers run in isolation and only communicate with each other when necessary [72]–[75]. A container functions similarly to a very light virtual machine [73]. A virtual machine is a large-scale compute resource. A virtual machine is a clone of an operating system that runs on top of a hypervisor that runs on top of physical hardware, on top of which the application runs. This poses certain speed and performance problems, as well as challenges in an agile setting [73]. Due to using the operating system kernel docker containers are very light and can launch in a fraction of a second. As a result, many of them can be fit on a physical or virtual machine [73], [74].

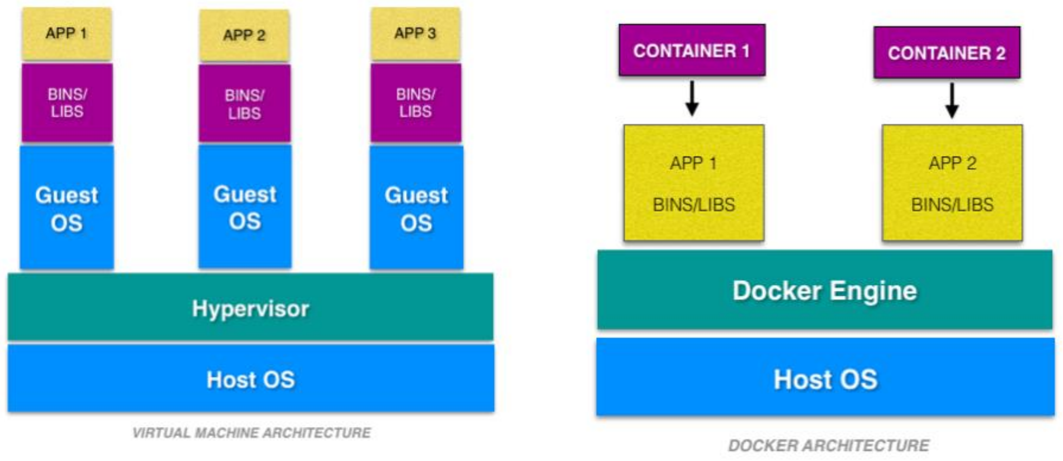


Figure 13: Vms vs Containers [76]

Docker architecture

Docker Client, Docker Daemon or Server, Docker Images, Docker Registries, and Docker Containers are the five key internal components of Docker. As mentioned in [72], [75], these components will be explained in detail in the following sections.

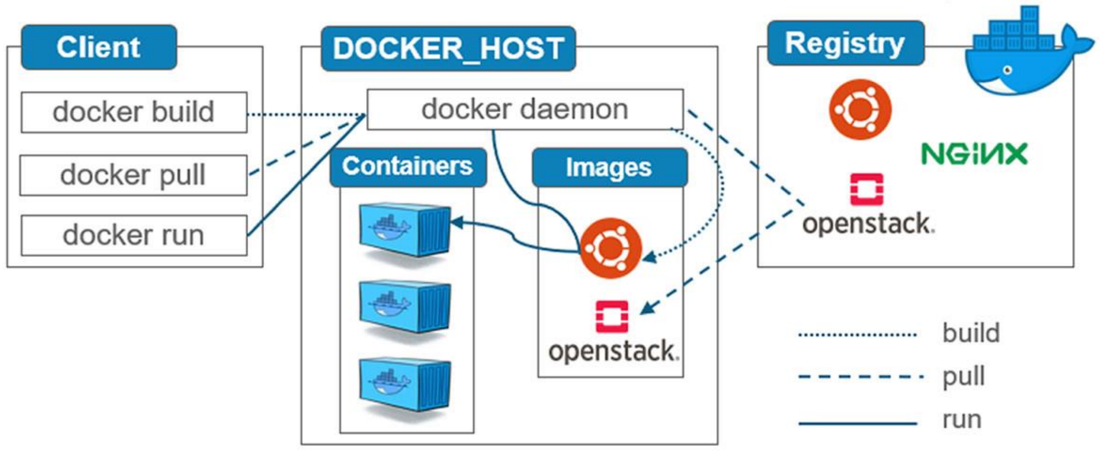


Figure 14: Docker architecture [77]

- Docker client: For many Docker users, the Docker client is the primary way of interacting with Docker. When users use docker run command, the docker client sends these commands to the Docker daemon, which executes them and returns a response. The Docker client can communicate with many daemons.
- Docker daemon: The Docker daemon or docker server is handling Docker objects such as images, containers and more by listening for API requests sent by the client. Docker servers can communicate with each other when needed.
- Docker registries are places where Docker images are stored. The registry that everyone uses by default is called docker hub. Creating private registries is also possible.
- Images are text-based templates that include instructions for creating and starting a Docker container. Most of the time, images are using other images as a base. For example, when users want to make a docker image that starts a Nodejs server they can use a linux image as their base image and then add the commands needed to install node as well as the necessary configuration data needed to start the server.
- Containers: A container is the result of executing the instructions contained in a docker image. Using commands, users can create, start, stop, move, or destroy a container. Users can attach local storage to a container, connect it to one or more virtual networks, or even create new docker images based on the current filesystem state of the container. By default, most containers run isolated from other containers and the host machine. However, users have control over how isolated a container is by controlling its network and storage.

5.1.2 Kubernetes

Kubernetes is an open-source orchestration solution for automating container management, placement, scaling, and routing that has recently gained popularity among developers and IT operations teams. The Cloud Native Computing Foundation now maintains it, which was previously developed by Google and submitted to Open Source in 2014. Kubernetes provides a uniform architecture for running distributed systems, ensuring that development teams have consistent, immutable infrastructure across all projects [78]–[80]. Kubernetes offers the ability to deploy multiple pods across actual servers, allowing an application to scale out with a dynamically changing workload. Each pod can manage one or multiple Docker containers, each of which can access the pod's services (such as the file system and I/O) [78], [79], including self-healing, which means that pods that fail are automatically restarted [81]. Kubernetes also manages communication between all pods by providing "services," which are means to communicate with pods regardless of where they are in the cluster or if they are replaced [81], [82]. Secrets, which are variables that securely store and communicate values across all pods, can be created using Kubernetes [81]. One of the benefits of Kubernetes is that it allows for horizontal scaling, which means that instead of increasing the resources for each pod, which can be difficult in some cases (for example, when using older machines), we can instead deploy multiple light instances of the same pod across multiple machines [81]. The usage of an ingress, which is a form of service that distributes traffic among pods carrying the same image, can enhance this feature even more.

5.1.3 React JS

Facebook created ReactJS, a JavaScript toolkit for creating modular user interfaces. React makes creating interactive UIs simpler. Component-based development is promoted by React, which means creating encapsulated components that maintain their own state and then composing them to create complicated UIs. Each component can have its own state, which can be updated efficiently when it changes [83]. According to [84], [85] some of the advantages of using react are:

- React delivers a significantly more performant DOM, which improves the overall app performance. It does not interact directly with the browser's DOM, instead relying on the virtual DOM. It works in a very straightforward manner. The diff() algorithm is used to compare the virtual and actual DOMs, and only the node modifications are propagated back into the document object model tree.
- ReactJS has a simple and non-complex learning curve, allowing users to quickly become comfortable with the framework. The learning curve is quite simple and allows one to progress without difficulty.
- React development is done with JSX. JSX is comparable to xml, but it also includes JavaScript, making it easier to learn [86].
- ReactJS is mostly known for its excellent performance. This along with other features helps distinguish react from the other frameworks that are currently in the market. The virtual DOM component of the library, as described above, is primarily responsible for the framework's great efficiency.

5.1.4 JavaScript & Typescript

JavaScript is a programming language that allows web developers to add advanced functionality to their pages. When a web website does more than display static material for users to look at, such as when it displays interactive components such as interactive maps, animated 2D/3D interactive visuals, scrolling animations, and so on. It's likely that JavaScript is involved [87]. JavaScript is used to interact with and change the dom, which contains all of a website's CSS and html parts, allowing for intricate interactions [88], [89]. The TypeScript language is a typed superset of JavaScript that is compiled to plain JavaScript, with optional types as one of the key improvements. Optional types' main goal is to combine the flexibility of a dynamically typed language with the benefits of statically typed languages, such as the compiler's ability to quickly discover obvious coding errors and editors' capacity to give code completion. Types are also useful for documentation, as they give programmers a streamlined language to document APIs [90], [91].

5.1.5 Node JS

Node.js is a JavaScript environment based on the "V8" engine developed by Google. Since node is based on "V8" that means they are mostly written in C++ making them performant and highly efficient when it comes to resource management. However, while V8 primarily focuses on JavaScript running in the browser, Node aims to allow creating long running processes running in servers. Compared to other environment that use multithreading for concurrent execution Node processes are single threaded. To provide concurrent execution, it uses an asynchronous I/O event mechanism. Because Node supports the event mechanism at a low level such as the language level, it differs from most eventing solutions of other programming languages, which are usually libraries [92], [93]. To handle a variety of events, NodeJS employs an event-loop, which runs asynchronously on a single thread and can expand to millions of concurrent users. NodeJS avoids wasting clock cycles by using a non-blocking event loop to handle all accessible events-requests. The non-blocking event loop in the single-threaded JavaScript architecture takes almost all events-requests and has an asynchronous design that employs callbacks to prevent wasting clock cycles and has enough memory to handle them [93].

Node also has some built in function however it also offers a package manager called Node Package Manager (NPM). The three types of Node modules are core modules, third-party modules, and local modules. All of them can be installed with the Node CLI, either during NodeJS initialization for the core scripts or later for third-party modules from the NPM repository. Local modules are generated by a user, who normally imports them locally before distributing them via the NPM repository [93].

5.1.6 Mongo dB

MongoDB is a database management system that may be used for both small projects with simple databases and products and applications with extensive databases. Bosch, Cisco, Toyota, and others are among the companies that use Mongo dB [94], [95]. A set of collections is stored in the MongoDB database. A collection, unlike tables, does not require a preset schema and stores data as BSON documents. A document is a group of fields that can be compared to a database record. It can include simple fields like integers and strings, as well as more complicated structures like arrays and even a full document. Each document has a unique ID field that can be used as a primary key, and each collection can hold multiple types of documents [94], [96]. Because each document can have a unique structure, using a non-relational database like MongoDB for development provides a lot of freedom. In a relational database, for example, it is advised that arrays (such as an array of several user emails) be stored in a separate table. However, in a non-relational database, numerous emails can be stored as an array on the same document [96]. The document format used by Mongo dB is based on JSON, making it appropriate for storing arbitrary data structures [96].

5.1.7 Express JS

Express is a web app framework with a simple and versatile feature set for web and mobile apps [97]. Express is based on the node JS http module [98]. Express aims to make creating APIs easier by incorporating conveniences into the node http server, which can be quite complex [99]. The following are some of the conveniences that express provides: (a) simpler http body parsing, (b) simpler cookie parsing, (c) simpler session management, and (d) simpler router organization [98]. Without express, all these capabilities would need a lot of difficult code to achieve, but with express, they can be done in a few lines of code. For example, instead of designing a request handler that sends a JPEG file, programmers can utilize express's sendFile function [98], [99]. Express encourages the usage of smaller reusable functions that may be built by developers or imported from third-party libraries, rather than developing a large request handler. These modules are called middlewares [99]. When it comes to scaling projects, Express is agnostic, although it does have a useful feature called "sub applications" which allows routes to be split into numerous files. The code is more maintainable and understandable when monolithic request handlers are split down into smaller handlers [99]. Finally, Express provides an MVC-like structure [98].

5.1.8 HTML & CSS

Tim Berners-Lee, and others designed HTML in 1989. It is a language for creating web page documents that stands for Hyper Text Markup Language. Unlike programming languages Html is a markup language, meaning it is a method for identifying and defining the various components of a document, such as paragraphs, lists, and headings as well as how they are processed and displayed. It is not a programming language. Tags and attributes are used in HTML to do this. Tags are used to indicate the beginning of an HTML element and are often enclosed in angle brackets. <h1> is an example of a tag. Attributes are used to include extra information and are represented by an opening tag with extra information inside. <img src="mydog.jpg" alt="A photo of my dog."> is an example of an attribute [100], [101]. While HTML is used to manage the content of a web page CSS describes how this content should look. CSS (Cascading Style Sheets) is a stylesheet language for managing and changing the way an HTML or XML document looks and feels. CSS specifies how elements should appear on a screen. That implies CSS handles fonts, colors, line spacing, page layout, and so on [100], [102].

5.2 Software implementation

The implementation of the platform's primary components and functionality is detailed in this section. To explain how these features were implemented, code excerpts and graphs will be presented.

5.2.1 Execute challenges implementation

The execute challenges service is one of the most significant services in the app since it handles one of the app's fundamental functionalities, which is executing user-submitted code and grading it according to professor-set standards. The service comprises a backend application and a database that stores tests for each challenge, such as input output tests, expected design patterns, and expected structure. When the service receives code from a user, it formats it according to a grading template. This template varies depending on the challenge's language, but the process is the same regardless of the language. First, the user-submitted code is appended to a new text file, and further templates are appended depending on the tests specified. The output-checking template runs the functions supplied by professors and verifies that the output satisfies their expectations. If affirmative, the totalTestsPassed variable is incremented by one. A pseudocode example of this method may be found in figure 15 It's worth noting that test.input is a function call with the professor's arguments. If the needed function has been appropriately implemented, the value returned by this method when called with certain arguments should match the expected output.

```
totalTestsPassed = 0
for (test in testsList)
        if (test.input === test.output)
                totalTestsPassed = totalTestsPassed+1
return totalTestsPassed
```

Figure 15: Pseudocode of output testing

This service also checks code structure and detects design patterns in addition to checking output. Because both structure checking and design pattern checking rely on the code structure, these two features are very similar in logic. When the professor requires either of these features, a function that detects code structure is appended to the head of the file containing the user's code. This function recognizes the code structure and returns it in a javascript-friendly format. Finally, as shown in figure 16 if the professor selects the code structure checking feature, the professor's code structure is compared to the output of the structure detecting function.

```
const checkStructure = (classes: classDiagram[], expectedClasses: classDiagram[])
=> {
    let found = 0;
    //for each expected class
    expectedClasses.map(targetClass => {
        //for each provided class
        classes.map((currentClass,index)=>{
            //check if class names match
            if(currentClass.className===targetClass.className){
                // if yes check if classes implement the same interfaces
                if(compareArrays(currentClass.interfaces,targetClass.interfaces))
{
                    //if yes check if classes contain the same modifiers (e.g.
public static)
                    if(compareArrays(currentClass.modifiers,targetClass.modifiers
)){
                        // if yes check if classes extend the same class
                        if(currentClass.superClass===targetClass.superClass){
                            // if yes check if provided class contains all the
methods of the expected class
                            if(targetClass.methods.every(el=>currentClass.methods
.some(method=>method.name===el.name &&
compareArrays(method.modifiers,el.modifiers) && method.returnType===el.returnType
&& compareArrays(method.parameters,el.parameters)))){
                                // if yes check if provided class contains all
the constructors of the expected class
                                if(targetClass.constructors.every(el=>currentClas
s.constructors.some(constructor=>
compareArrays(el.modifiers,constructor.modifiers)&&
compareArrays(el.parameters,constructor.parameters)))){
                                    // if yes check if provided class contains
all the fields of the expected class
                                    if(targetClass.fields.every(el=>currentClass.
fields.some(field=>compareArrays(el.modifiers,field.modifiers)&&el.name===field.n
ame&&el.type===field.type))){
                                        found++;
                                    }
                                }
                            }
                        }
                    }
                }
            }

        })
```

Figure 16: Class structure comparison function

The application assumes that the required structure is found if the user's code structure contains the required structure. The way design pattern detection works is fairly similar to this, with the exception that the developers specify the required structure for each design pattern. When a user submits code and the professor has specified a required design pattern for this exam, the structure of the code is compared to the structure defined by programmers for this design pattern. The design pattern is considered found if the structure matches. Following the completion of all of these tests, a final grade for the submitted code is determined. This grade takes into account how many output tests passed, whether the code structure fits the required structure if one was provided, and how many of the required design patterns were detected, if any. This grade is sent to the user, and if the user chooses to submit the assignment, the grade is sent via an event to the history service where the grade is stored. A flow diagram of all the above-mentioned service functionalities is shown in the figure below.

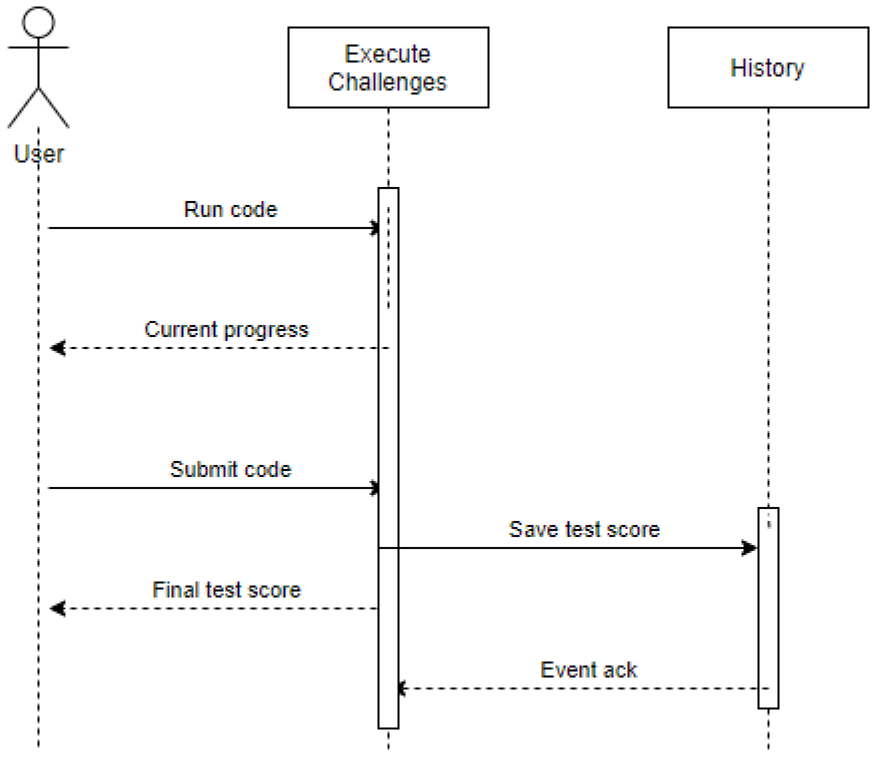


Figure 17: execute challenge sequence

Optimizing the code execution

Because code execution can take time, execution times were a major issue while building the execute challenge service. Initially, each test was compiled separately, however when dealing with several tests, the grading process could take several minutes, resulting in a poor user experience and a high server load. As a result, a more efficient method was chosen. Instead of many compiles, the entire grading process is now handled by a single compilation. This has greatly accelerated the process. A comparison of service performance before and after optimization is shown in the figure below.

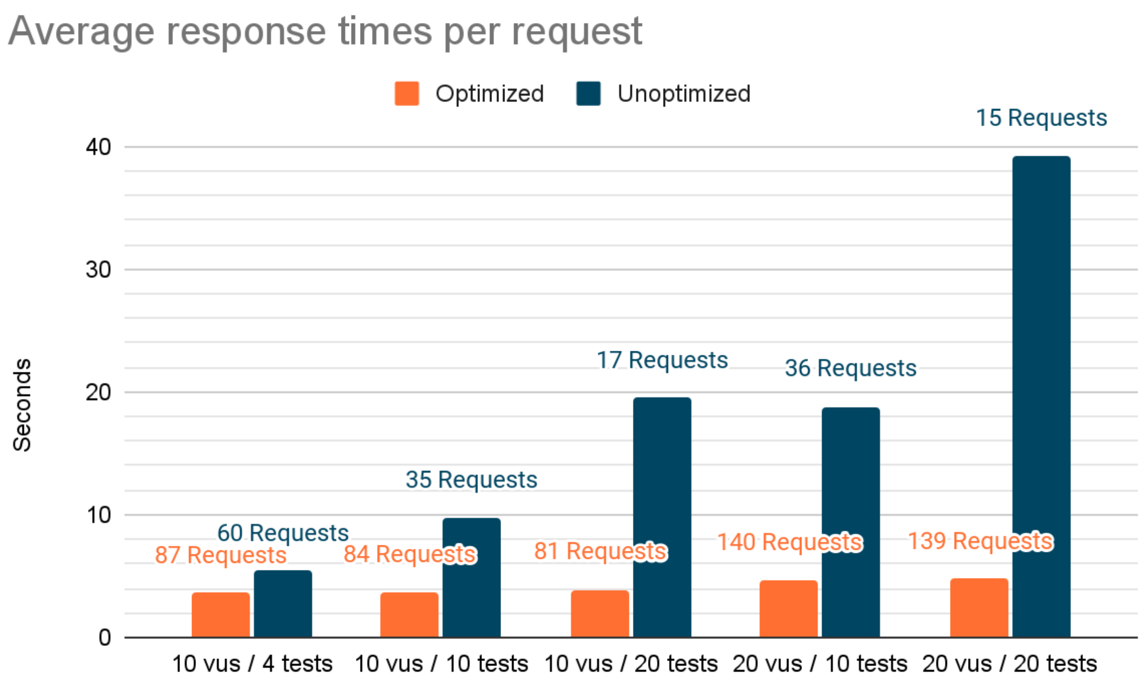


Figure 18: Execute challenges performance before and after optimization

In the figure above is presented the performance before and after optimization. In the vertical axis is the average response time in seconds. In the horizontal axis are the conditions of the tests. For example, 10 vus / 4 tests means there were 10 concurrent users each submitting an exam that contains 4 tests. Also, on top of each bar is shown the amount of requests that the server managed to respond to in a 35s period. From the above graph it is easy to grasp that the unoptimized version's performance was heavily affected by the number of tests. Meaning that if a test requires x number of seconds to be completed 2 tests would require 2x seconds. However, when all the tests are done in a single compilation the amount of tests does not affect the compilation time that much. Finally, in the graph is also presented the number of requests the servers managed to respond to in an about 35 sec period. From the graph it is obvious that in the unoptimized version the servers struggled when dealing with the heavy load and the number of tests affected the amount of total fulfilled requests. However, this wasn't the case for the optimized version who managed to fulfill almost the same amount of requests despite the number of tests. In one case not shown in the graph above when the system was tested with 10 users and 50 tests the unoptimized version didn't respond to a single request in the 35 sec period, meanwhile the optimized version handled 73 requests with an average 4.21 second response time.

5.2.2 Challenge moderation implementation

While implementing the challenges service the quality of the application content was a huge concern. Students shouldn't be exposed to inappropriate content and challenges have to have good content and valid tests. As a result, the creation of a moderation mechanism was deemed necessary. In order to implement the moderation of the challenges a new service had to be created. This service receives requests submitted by users from the challenges service and stores them until an admin approves or declines them. As shown in the figure below these requests only concern public challenges since they are the ones appearing in the front page. Requests can be for the creation of new challenges, the modification of existing challenges, or the deletion of existing challenges.

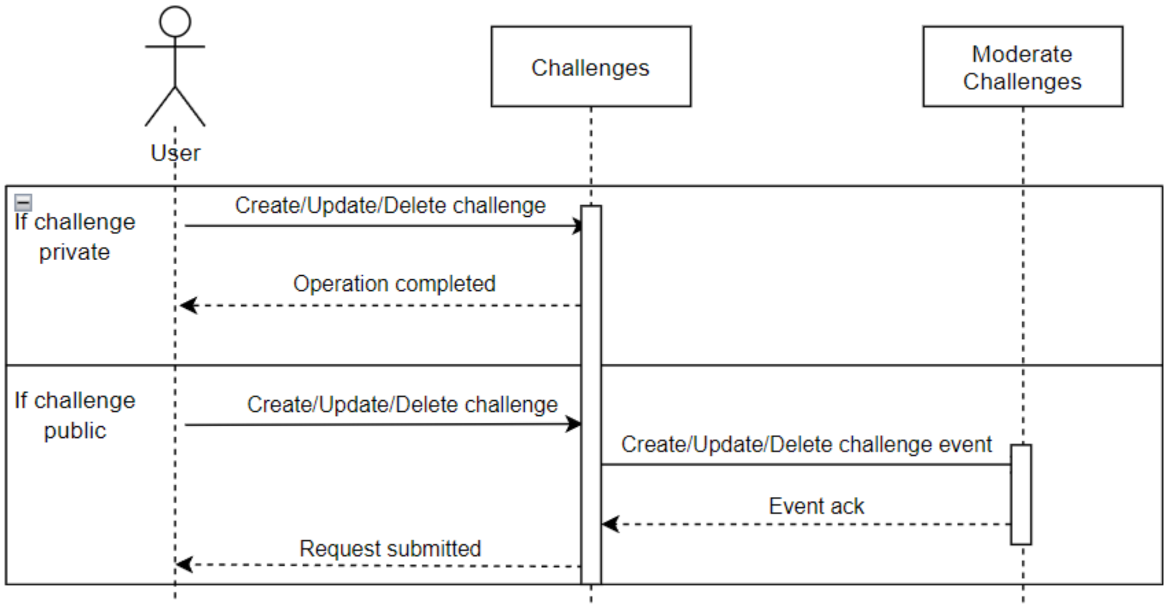


Figure 19: Challenge operation sequence diagram

When users attempt to create, update, or delete a private challenge, the operations are processed instantaneously, as illustrated above. When users attempt the same operation on public challenges, however, their requests are sent to the moderate challenge service. Admins can approve or decline requests after they have been stored in the moderation service. As depicted in the diagram 18 Admins receive pending requests from the moderate challenges service, and if they approve them, they are forwarded to the challenges service, or they are deleted if they are denied.

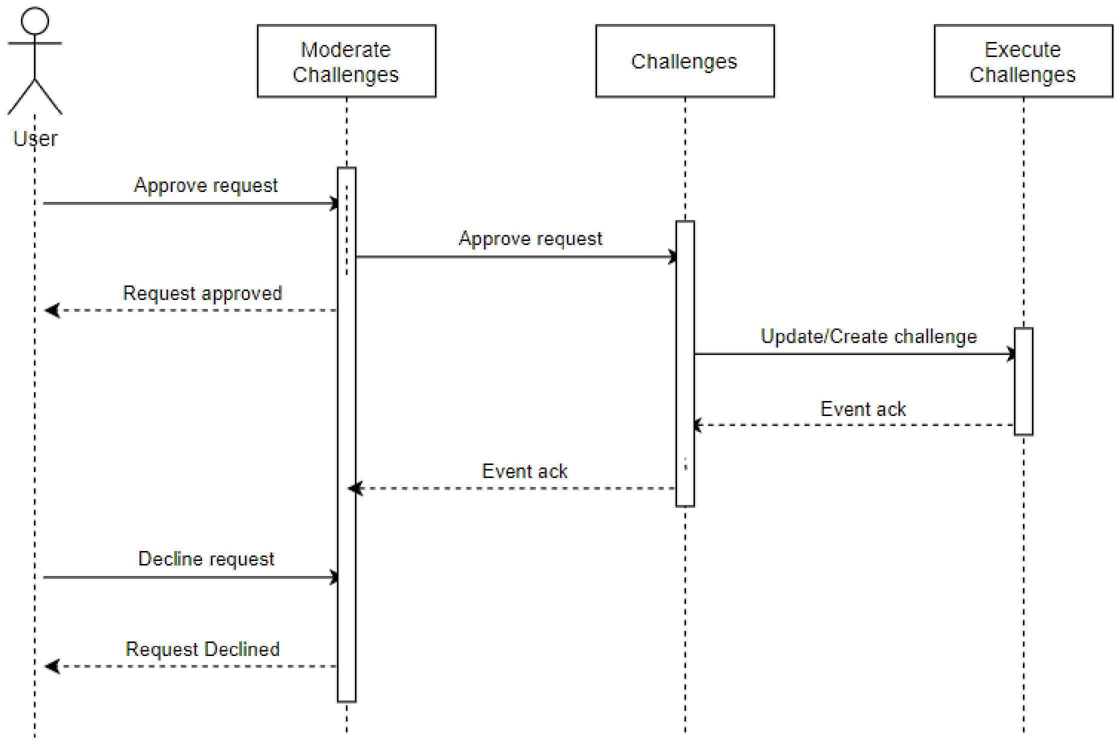
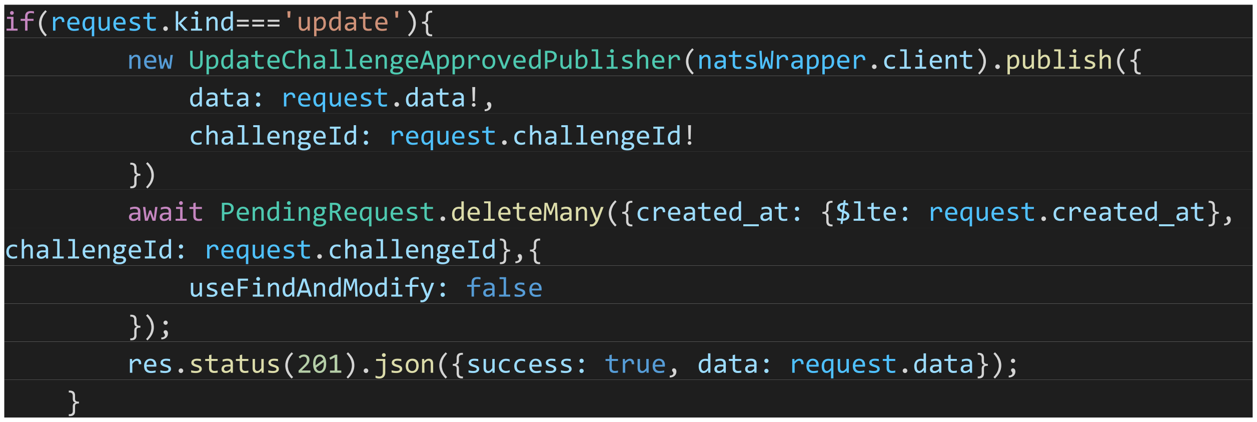


Figure 20: Challenge moderation sequence diagram

Request versioning system

The possibility of several requests for a single challenge was one of the issues encountered during the development of the moderation service. This was problematic since users may submit many edit requests for a single challenge, forcing administrators to go through multiple requests and approve only the best while rejecting the rest. It was also critical to allow users to override existing requests in the event that they made a mistake. As a result, a versioning system was implemented using stacks. All requests for each challenge are stored in stacks, and when a modification request is made, it is added to the stack for that challenge. When users attempt to modify a challenge, they are prompted to use the most recent pending request (at the top of the stack) as a template for the edit form. If users choose to, they will be able to make changes to the most recent request, if they do not, the current challenge will be utilized as a template. The new request is placed to the top of the stack when a modification has been made. A pending request can be canceled at any time, resulting in the stack's head being popped. It's worth noting that if a delete request is at the top of the stack, users are prompted to cancel it before submitting an edit request. Admins consider the most recent request while reviewing a pending request. If they approve the request, all requests in the stack from this request and down are removed, which means that if a user makes another request while admins are reviewing one, it will not be lost. In figure 19 is displayed the process of approving an update request as implemented.



```
if(request.kind==='update'){
        new UpdateChallengeApprovedPublisher(natsWrapper.client).publish({
            data: request.data!,
            challengeId: request.challengeId!
        })
        await PendingRequest.deleteMany({created_at: {$lte: request.created_at},
challengeId: request.challengeId},{
            useFindAndModify: false
        });
        res.status(201).json({success: true, data: request.data});
    }
```

Figure 21: Approving an update request

If the request is declined, then the head of the stack is popped, and admins can review the previous pending request for this challenge which is now at the top of the stack. The figure below provides a visualization of the process in order to better understand how it works.

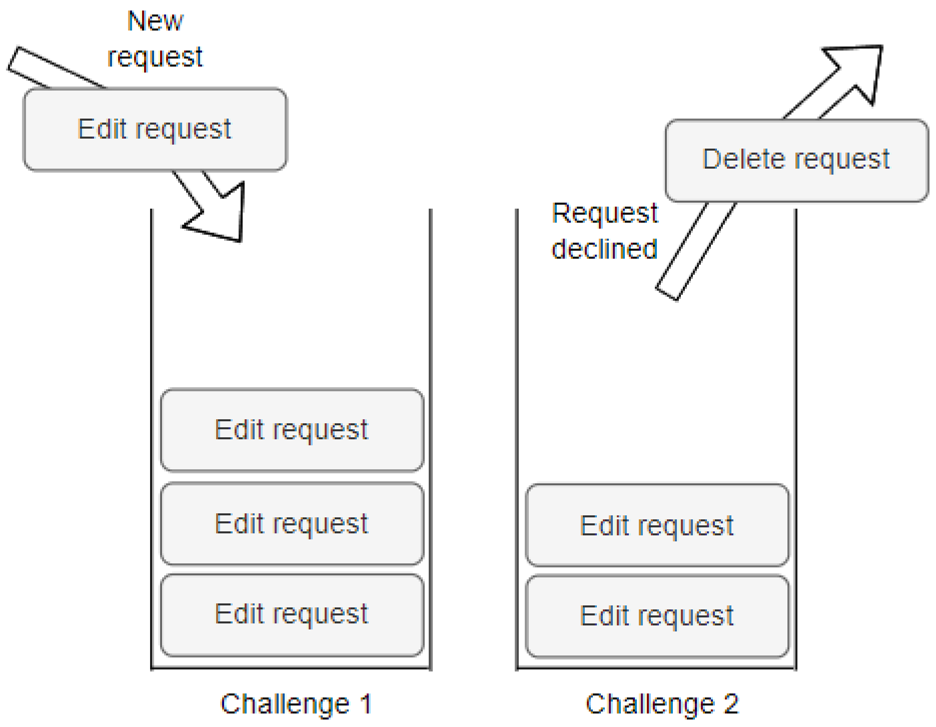
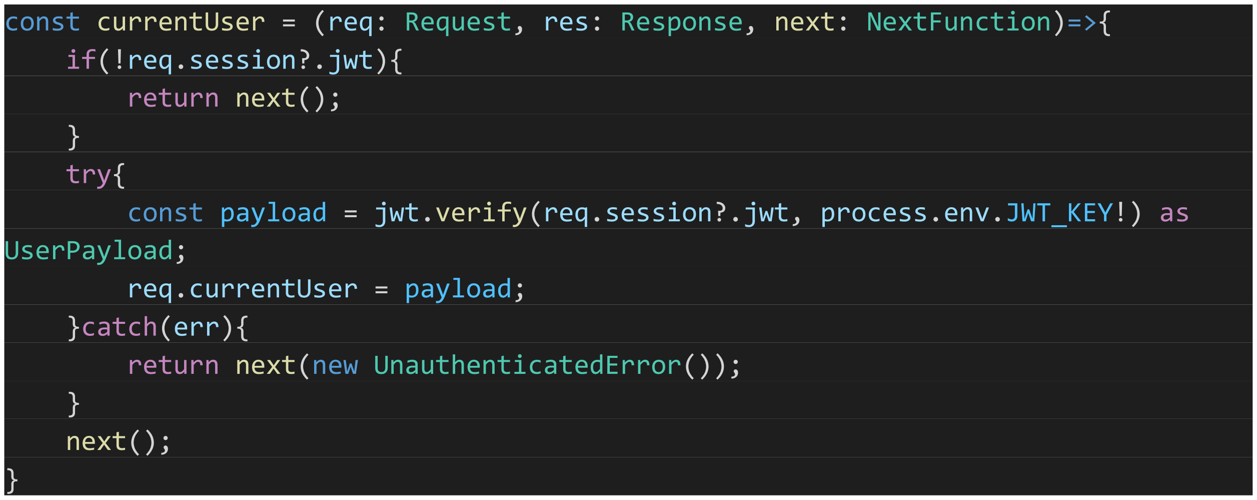


Figure 22: Pending requests stack

5.2.3 Authentication implementation

Authentication and authorization can be difficult on a microservices architecture especially when trying to keep all services independent of one another. One solution would be to create an authentication service that handles all the authentication and authorization, however if all services depend on one service, then all services would stop functioning if that service fails. Authentication is commonly done through a JWT in monolithic apps. A JWT is an open standard that specifies a method for securely transferring data between two parties. The JWT token is a signed JSON object with a set of claims that the receiver can use to verify the sender's identity. When a server receives a request, the JWT is usually included in the header or as a cookie, which is then decoded using the secret key and the data stored inside is used to authenticate the user. This can be handled by the authentication service in a microservices design, but as previously said, if this service fails, the entire app would fail. As a result, a stateless approach was selected. In order to login or register in this app, users must use the authentication service. When users authenticate, they obtain their encrypted JWT key; now every time a user tries to communicate with a service instead of relying on the auth service to decode their key, each service can utilize a method contained in the common module.



```
const currentUser = (req: Request, res: Response, next: NextFunction)=>{
    if(!req.session?.jwt){
        return next();
    }
    try{
        const payload = jwt.verify(req.session?.jwt, process.env.JWT_KEY!) as
UserPayload;
        req.currentUser = payload;
    }catch(err){
        return next(new UnauthenticatedError());
    }
    next();
}
```

Figure 23: Get current user method as implemented inside Eurytus common module

As shown in the figure above this method decodes the JWT and uses its content to authenticate the user. A key is necessary to encode and decode the JWT to keep this key safe in the event that a service is compromised, the key is kept as a secret inside the cluster. By storing the secret key in the cluster, all services can access it through the cluster's API. There is less chance of the Secret (and its data) being leaked because it is held apart from the services that use it. This protects the key and ensures that it is only used when necessary; it also makes it easier to update because it only needs to be changed once rather than several times for each service. This method decentralizes the authentication process and allows services to securely authenticate users without relying on each other. The only disadvantage of this strategy is that because the services do not communicate directly with the authentication service, if a user is blocked or banned, other services will be unaware because the user's JWT remains valid. This can be solved by using shorter token expiration durations and refresh tokens; as a result, tokens will expire and renew every few minutes, and banned users will be unable to reauthenticate; but, until their token expires, this solution will grant blocked users' access. Another reliable option is to employ a block event; when a user is banned, all services are immediately alerted via an event, allowing users to be blocked immediately.

5.2.4 Event communication implementation

An event system is used to communicate across microservices, as discussed previously. An event bus was used to implement this system. The publish-subscribe pattern is used by the event bus. It allows consumers to subscribe to specific channels containing messages of certain types and allows producers to publish messages to specific channels. When a producer sends out a message of a given type, it is stored in the event bus and delivered to any services that have subscribed to that message type. When a subscriber receives an event, it sends an "ack" message to the event bus, indicating that the event has been received. If the event bus does not receive an "ack" message, the message will be published until it does, ensuring that no event is lost. Even if a service fails then the event bus will store and deliver any missed events whenever the service is restored. The event bus used in this app also distributes events among copies of the same service, thus if there are two copies of the challenge service, events will be delivered in a round robin fashion.

Concurrency control

One of the issues encountered during the event bus's deployment was that events had to be handled in a specified order. When trying to update information on an entry that is held in different databases across multiple services, it becomes clear why this is a huge problem. For example, in this application, the challenges service stores information about a challenge, whereas the execute challenges service stores information about the tests that are part of that challenge. When a challenge in the challenges service is updated, an update event is sent to the execute challenges service, causing the challenge to be updated as well. If the challenge is updated numerous times quickly, the events may not be received in order for a variety of reasons, resulting in a different version of the challenge in these services. As a result, a concurrency control mechanism had to be put in place. The selected mechanism is very simple, every entry has a version number that rises each time it is updated. When an event is published, the version is also included.

```
export class UpdateChallengeListener extends Listener<UpdateChallengeEventData>{
    subject: Subjects.UpdateChallenge = Subjects.UpdateChallenge;
    QueueGroup = 'executechallenge-service'
    async onMessage(data: UpdateChallengeEventData["data"], msg: Message){
        try{
            let challenge = await Challenge.findOneAndUpdate({_id: data.id,
version: data.version-1}, data,{
                new: true,
                runValidators: true,
                useFindAndModify: false
            })
            await challenge?.save();
            msg.ack();
        }catch(err){
            console.log(err)
        }
    }
}
```

Figure 24: Version checking when receiving an update challenge event

As shown in figure 22 when an update event is received, the version of its contents is compared to the version of the database entry with the same id; if the database entry's version number is one version lower than the event's contents, the entry is updated, its version is increased, and the event is "acked." If the versions do not match, the event will not be acknowledged, and the event bus will continue to publish the event until the service has received all the missed update events in the proper order.

6 Results / demo

The final result of the application, as designed and developed, will be presented in this chapter. Starting with the authentication page, the presentation will follow the flow of pages that users will see when using the app. Each section discusses a certain page and the elements that a user must be familiar with in order to fully utilize the system's capabilities. In the Eurytus application, there are two types of users: students and teachers, as well as administrators who manage the application. Each role has access to different features and pages, which are detailed in the subsections below.

6.1 Auth page demo

The authentication page is used by all unauthenticated users in order to access the main features of the app. Existing users can login using their credentials and new users can create a new account. Users who successfully authenticate get redirected to the challenges page.

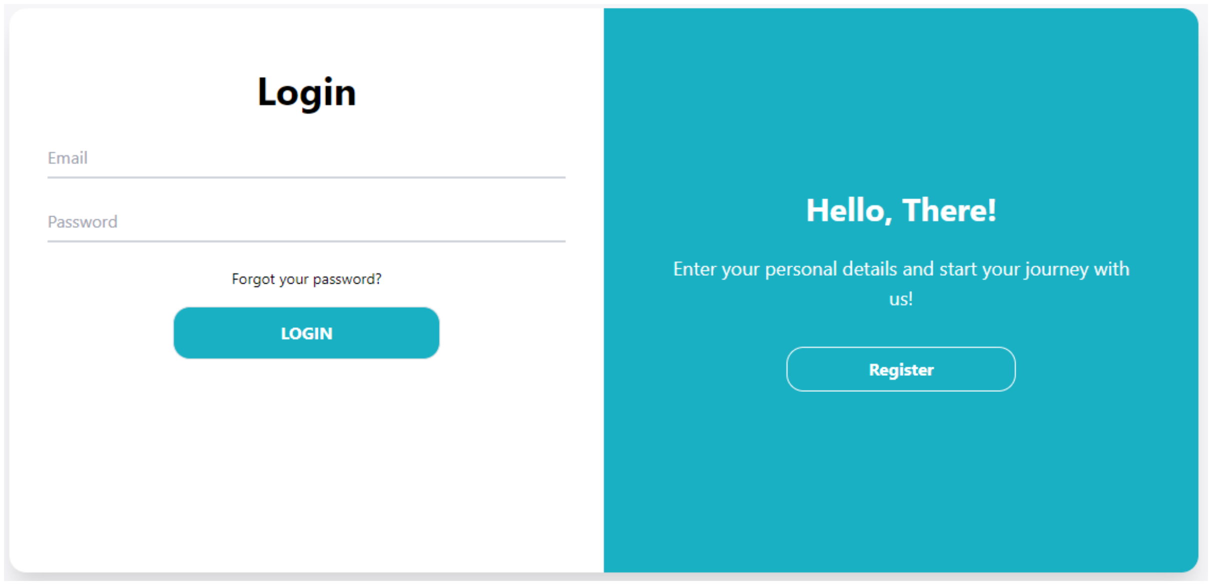
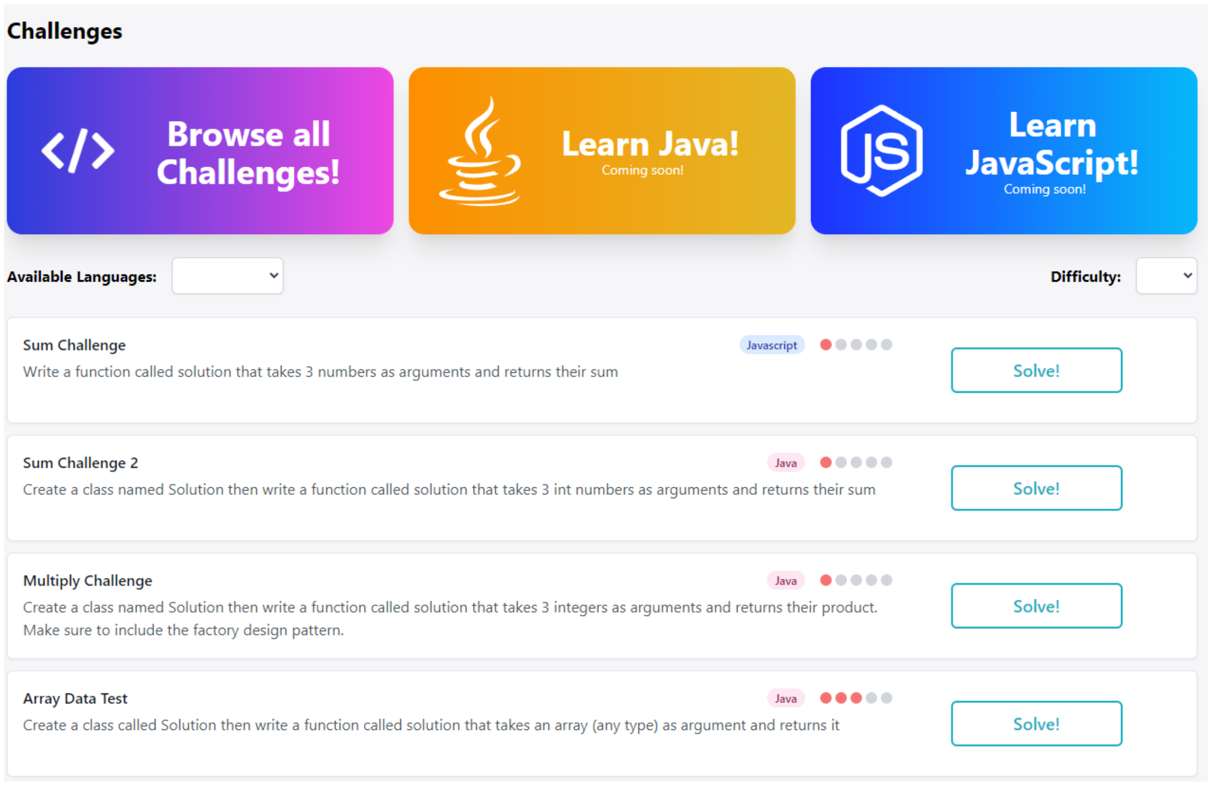


Figure 25: Authentication page demo

6.2 Challenges page demo

The challenges page is the home page for authenticated users. In this page users can browse all the public challenges and join the ones they might be interested in. Users can also filter the challenges based on difficulty and language to find challenges that better fit their needs.



Sum Challenge
Write a function called solution that takes 3 numbers as arguments and returns their sum

Sum Challenge 2
Create a class named Solution then write a function called solution that takes 3 int numbers as arguments and returns their sum

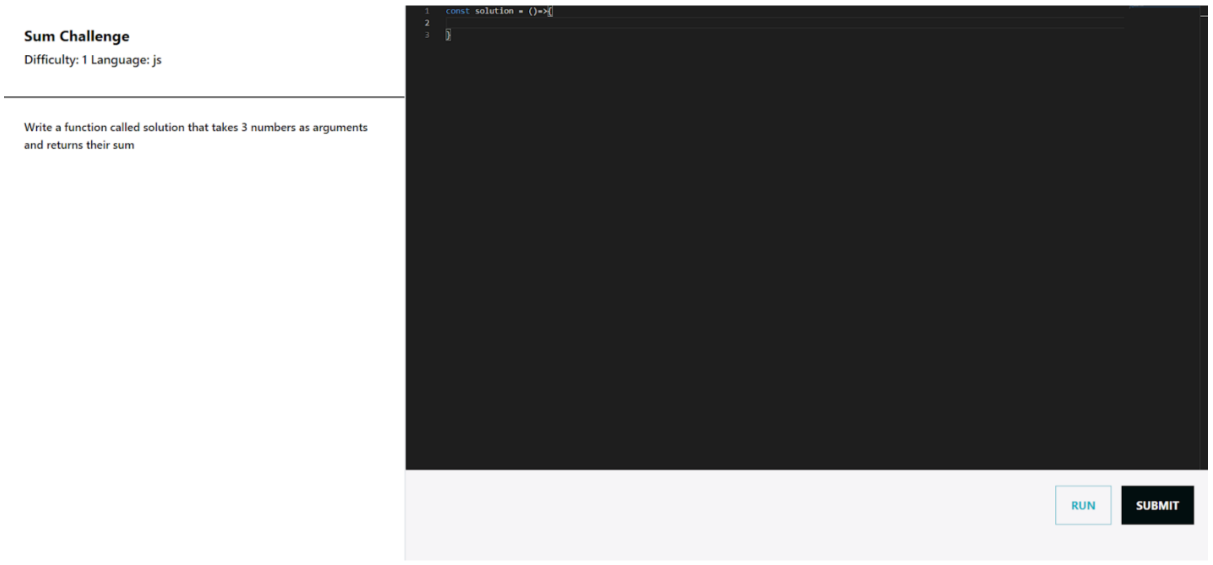
Multiply Challenge
Create a class named Solution then write a function called solution that takes 3 integers as arguments and returns their product. Make sure to include the factory design pattern.

Array Data Test
Create a class called Solution then write a function called solution that takes an array (any type) as argument and returns it

Figure 26: Challenges page demo

6.3 Solve challenge page demo

When users join a challenge, they get redirected to the solve challenge page. Users can join challenges from the challenges page as shown above, by using a URL provided by a professor or from the join challenge page. On the left of the solve challenge page users are given a description of the problem as provided by the professor when creating the challenge. On the right of the page there is an IDE where users can write their solutions. Under the IDE users can run their solution to check their progress or submit their code when they feel ready.



Sum Challenge

Difficulty: 1 Language: js

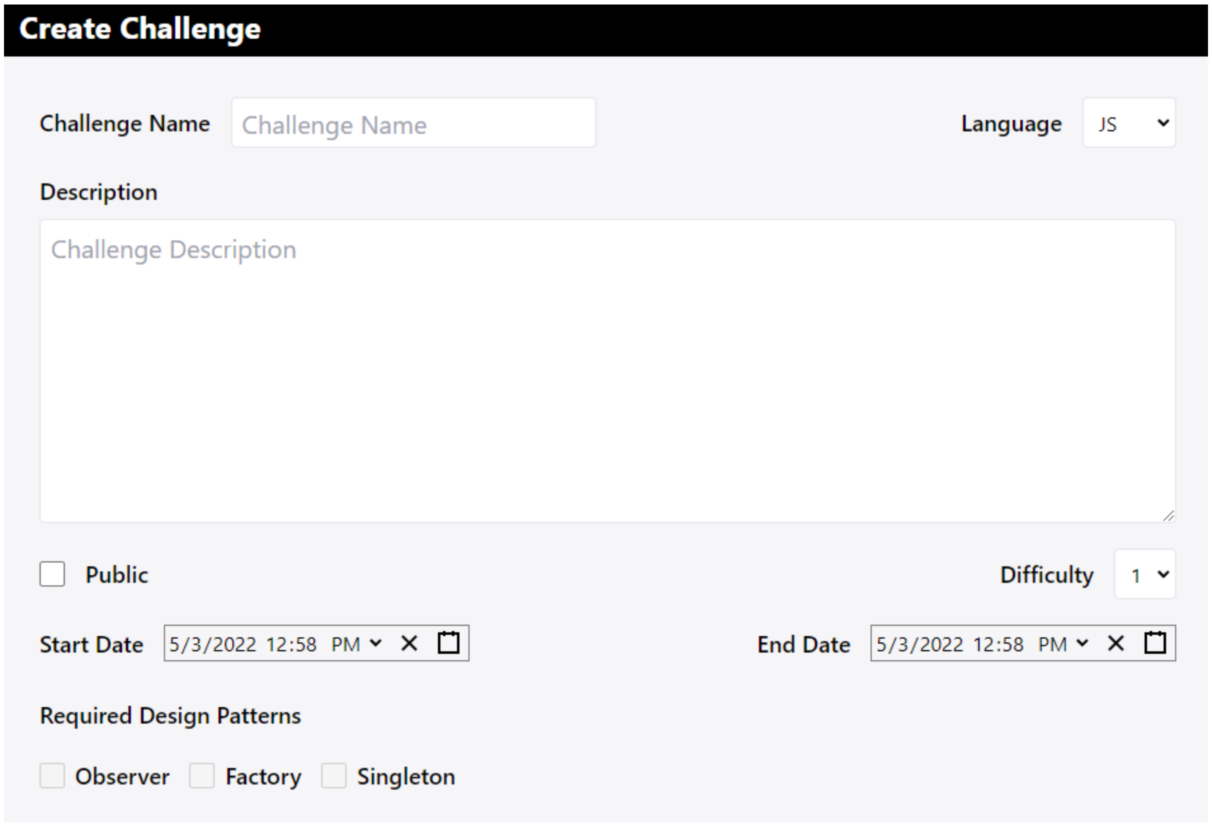
Write a function called solution that takes 3 numbers as arguments and returns their sum

```
const solution = ()=>{

}
```

Figure 27: Solve challenge page demo

6.4 Create challenge page demo

The create challenge page is a page where users can create new public or private challenges. As shown in the figure below in order to create a new challenge professors have to fill a form with the challenge details like name description etc. If the challenge is private then the start and expiration date for the exam can also be specified. For java challenges professors can also select required design patterns using this form.


Create Challenge

Challenge Name: Challenge Name

Language: JS

Description

Challenge Description

Public

Difficulty: 1

Start Date: 5/3/2022 12:58 PM

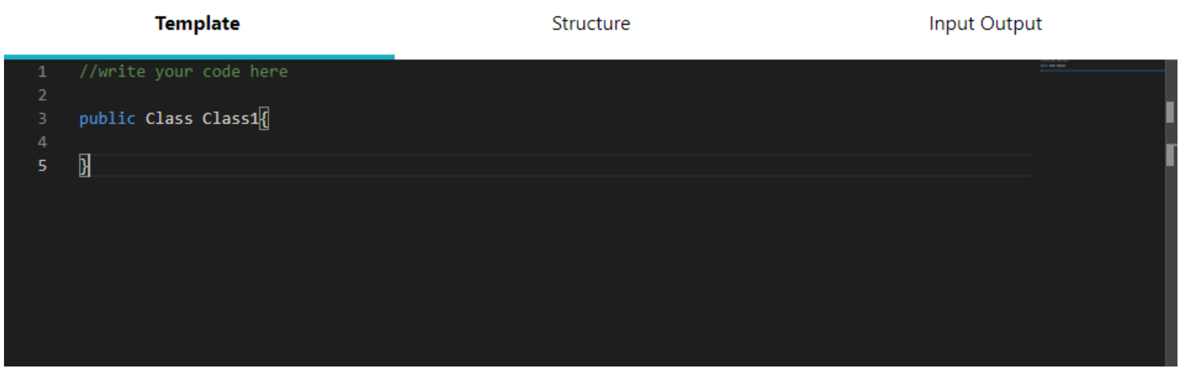
End Date: 5/3/2022 12:58 PM

Required Design Patterns

Observer Factory Singleton

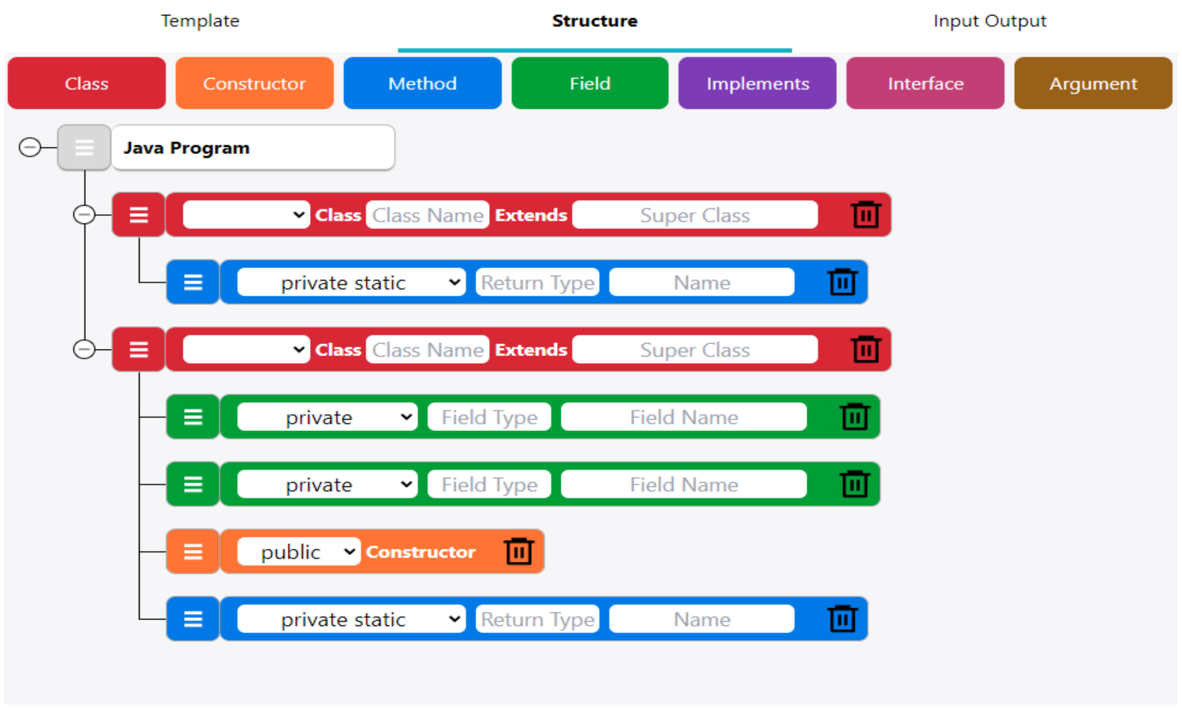
Figure 28: Challenge description demo

In this page professors can also specify the tests that will determine the students grade. As shown in the figure below professors are provided with 3 tabs. In the first tab professors can provide a starting template for their students so they can better guide them through the exam. The second tab is only available if the selected language is java, in this tab professors can specify a required structure for the solution using an interactive drag and drop tool developed specifically for this application. In the final tab professors have to specify the input and output tests that will run in order to grade the exam.



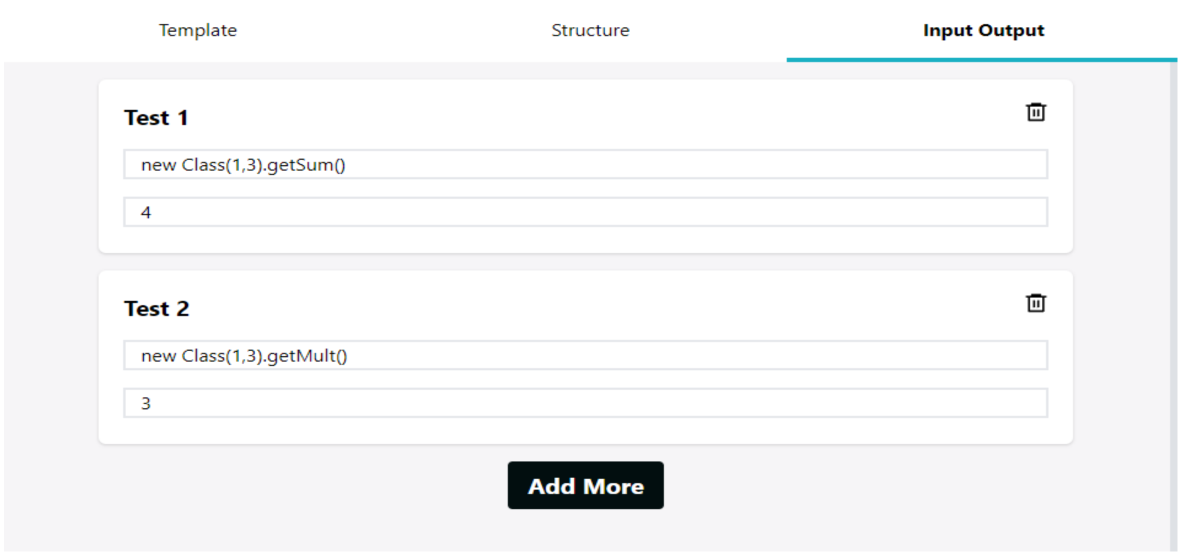
```
//write your code here

public Class Class1{

}
```

a) Challenge template demo



Template
Structure
Input Output
Class
Constructor
Method
Field
Implements
Interface
Argument
Java Program
Class Class Name Extends Super Class
private static Return Type Name
Class Class Name Extends Super Class
private Field Type Field Name
private Field Type Field Name
public Constructor
private static Return Type Name

b) Challenge required structure builder demo



Test 1

new Class(1,3).getSum()

4

Test 2

new Class(1,3).getMult()

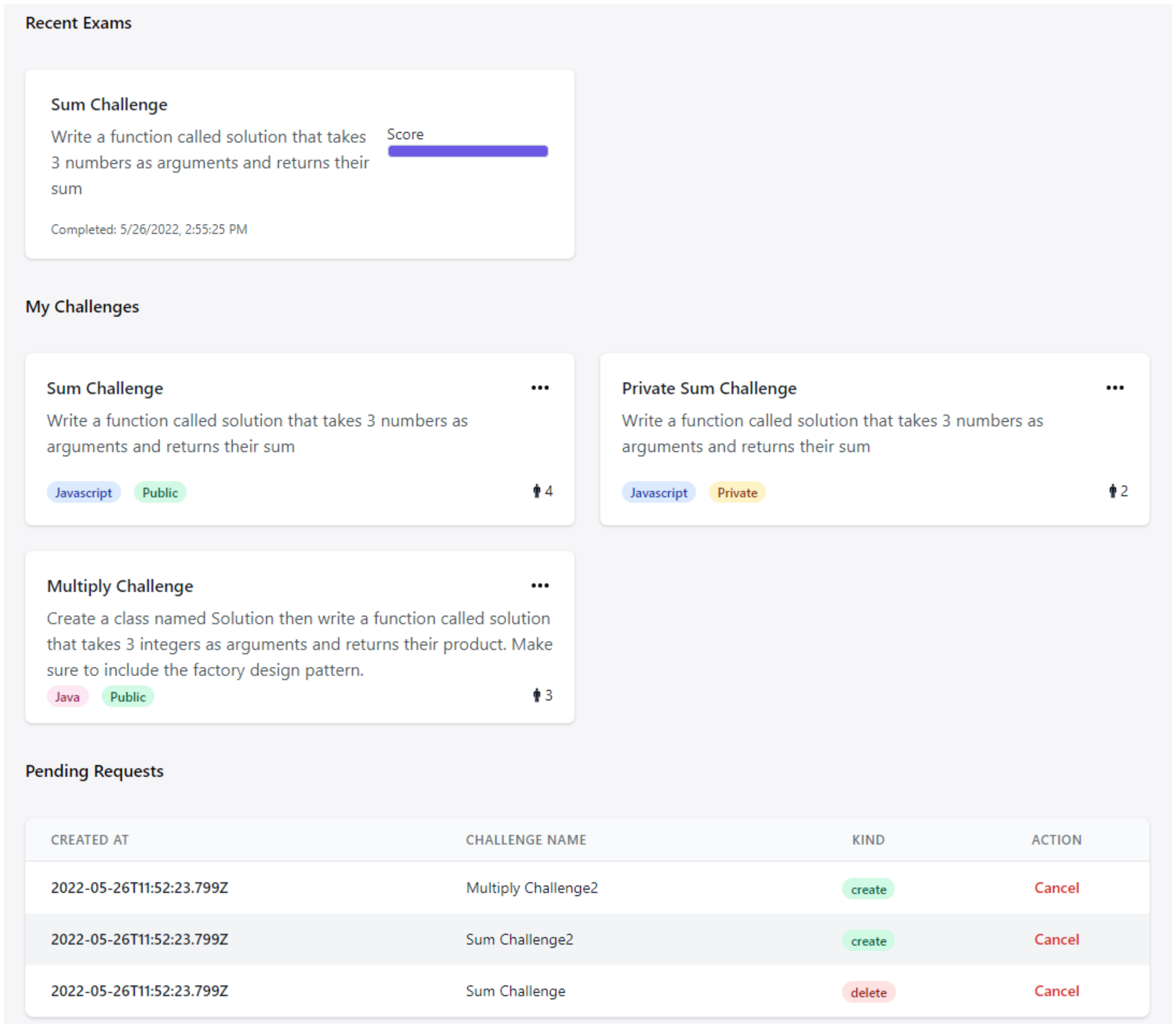
3

c) Challenge input output test creator demo

Figure 29: Challenge test creation demo

6.5 User profile page demo

The user profile page provides users with information about their profile and their activity. For students this page displays grades from previously completed challenges. For teachers this page displays all the exams created by the user and can also be used to manage the exams. Teachers can also manage their pending requests from this page.


Recent Exams

Sum Challenge
Write a function called solution that takes 3 numbers as arguments and returns their sum
Score
Completed: 5/26/2022, 2:55:25 PM

My Challenges

Sum Challenge
Write a function called solution that takes 3 numbers as arguments and returns their sum
Javascript Public 4

Private Sum Challenge
Write a function called solution that takes 3 numbers as arguments and returns their sum
Javascript Private 2

Multiply Challenge
Create a class named Solution then write a function called solution that takes 3 integers as arguments and returns their product. Make sure to include the factory design pattern.
Java Public 3

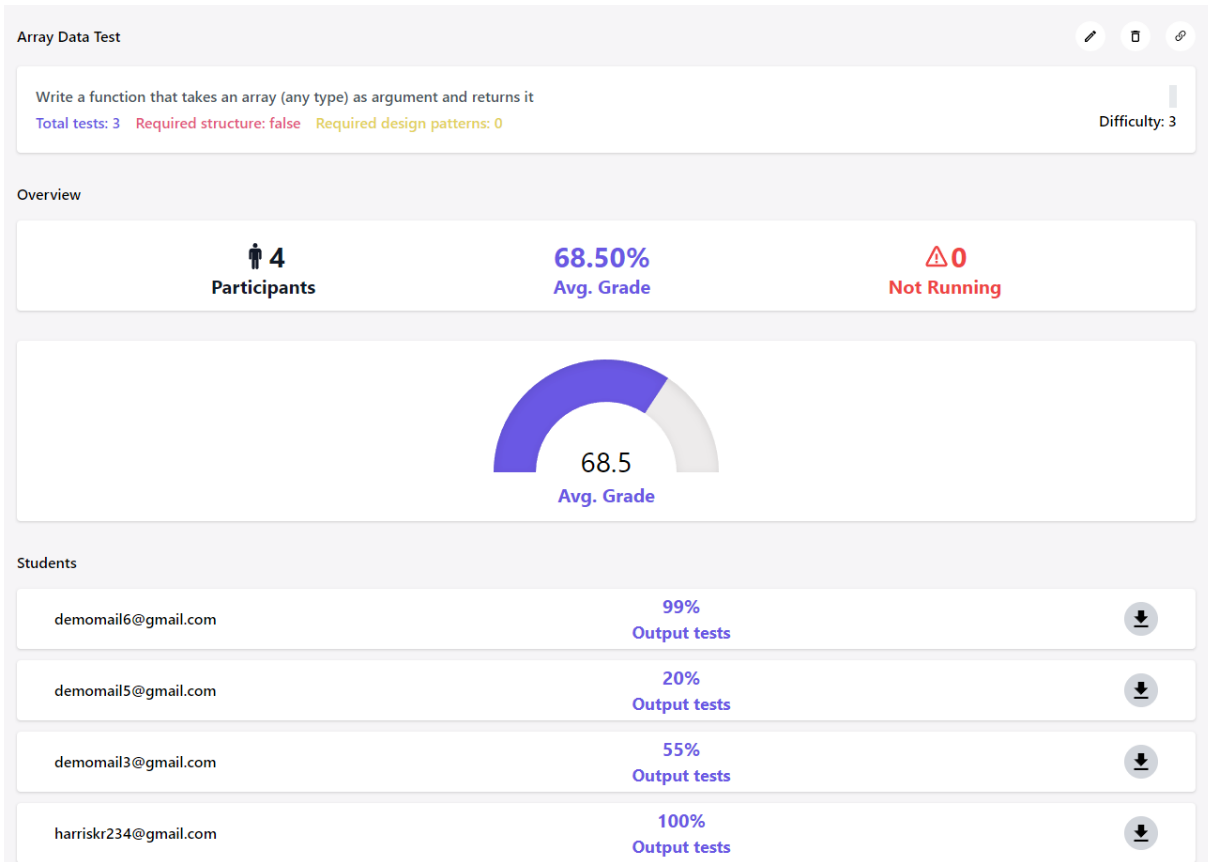
Pending Requests

CREATED AT	CHALLENGE NAME	KIND	ACTION
2022-05-26T11:52:23.799Z	Multiply Challenge2	create	Cancel
2022-05-26T11:52:23.799Z	Sum Challenge2	create	Cancel
2022-05-26T11:52:23.799Z	Sum Challenge	delete	Cancel

Figure 30: User profile page demo

6.6 Challenge statistics page

The challenge statistics page for a specific challenge is accessible only by the admins and the creator of the challenge. This page provides some info about the challenge details but also displays some statistics about the challenge. As shown in the figure below users can get a general idea of how participants performed but also check each user individually and download submitted solutions.


Array Data Test

Write a function that takes an array (any type) as argument and returns it

Total tests: 3 Required structure: false Required design patterns: 0

Difficulty: 3

Overview

4 Participants

68.50% Avg. Grade

0 Not Running

68.5 Avg. Grade

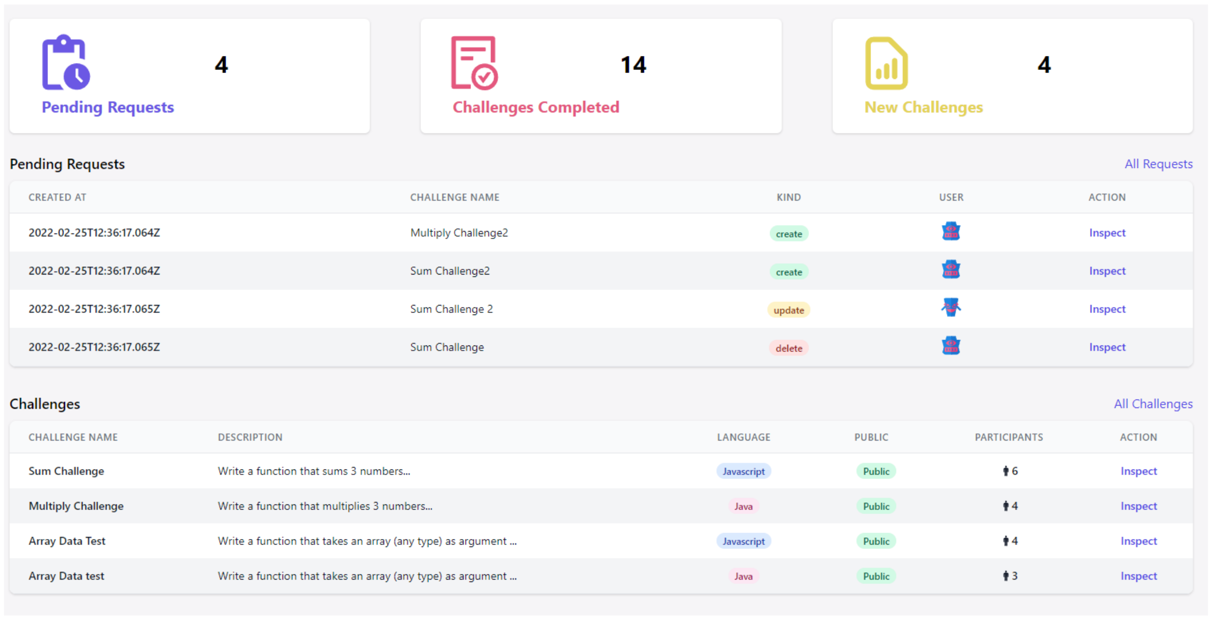
Students

demomail6@gmail.com	99% Output tests
demomail5@gmail.com	20% Output tests
demomail3@gmail.com	55% Output tests
harriskr234@gmail.com	100% Output tests

Figure 31: Challenge statistics page

6.7 Admin page demo

The admin page is only accessible by admins and can be used to view statistics about the application but also to manage challenges and pending requests submitted by users. Admins can view statistics for all the challenges in the app but also edit and delete them if needed. From this page admins can also examine all the pending requests submitted by users and approve or decline them.


4
Pending Requests

14
Challenges Completed

4
New Challenges

Pending Requests
All Requests

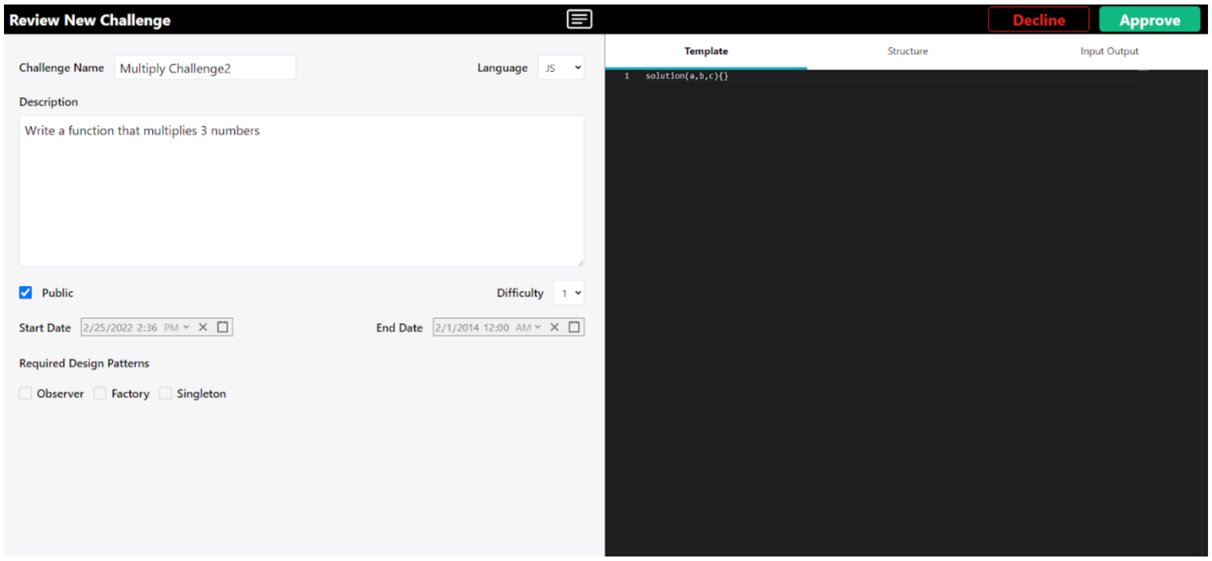
CREATED AT	CHALLENGE NAME	KIND	USER	ACTION
2022-02-25T12:36:17.064Z	Multiply Challenge2	create		Inspect
2022-02-25T12:36:17.064Z	Sum Challenge2	create		Inspect
2022-02-25T12:36:17.065Z	Sum Challenge 2	update		Inspect
2022-02-25T12:36:17.065Z	Sum Challenge	delete		Inspect

Challenges
All Challenges

CHALLENGE NAME	DESCRIPTION	LANGUAGE	PUBLIC	PARTICIPANTS	ACTION
Sum Challenge	Write a function that sums 3 numbers...	Javascript	Public	6	Inspect
Multiply Challenge	Write a function that multiplies 3 numbers...	Java	Public	4	Inspect
Array Data Test	Write a function that takes an array (any type) as argument ...	Javascript	Public	4	Inspect
Array Data test	Write a function that takes an array (any type) as argument ...	Java	Public	3	Inspect

Figure 32: Admin page demo

As shown in the figure below, when admins review a pending request regarding a public challenge they are provided with the form exactly as filed by the users. During the creation of the request users are asked to provide a short text explaining why they think admins should approve their request. After reviewing the details of the challenge as well as the justification for the request, admins can either approve or decline the request.


Review New Challenge
Decline
Approve

Challenge Name: Multiply Challenge2
Language: JS

Description
Write a function that multiplies 3 numbers

[✓] Public
Difficulty: 1

Start Date: 2/25/2022 2:36 PM
End Date: 2/1/2014 12:00 AM

Required Design Patterns
[ ] Observer [ ] Factory [ ] Singleton

Template | Structure | Input Output

```
solution(a,b,c){}
```

Figure 33: Review pending request demo

7 Conclusion & future work

The robust evolution of technology has impacted greatly the field of education resulting in the development of modern systems aiming to assist or even manage the whole education process. Examples of E-learning systems include asynchronous courses, synchronous online classrooms and many more.

The advancements in E-learning and the introduction of new pedagogical methods such as MOOCs, online learning platforms, and many others have considerably helped information technology education. Online coding learning platforms are a popular and low-cost choice for IT training. Such platforms offer a diverse set of challenges, with a particular emphasis on the algorithms and data structures that must be utilized while constructing a solution (program) to a problem presented by the student and must be graded to assess programming knowledge.

The goal of this thesis was to create an asynchronous code grading platform that would enable professors to teach code more efficiently while simultaneously assisting students in learning more effectively.

Eurytus, the proposed system, not only handles the assignment and grading of coding exercises, but also allows community members to create public programming challenges that anyone may participate in. It can be used by professors who need help grading examinations and assignments, as well as people who want to practice and improve their programming skills. The system was built with cutting-edge open-source technologies and a microservices architecture, making it highly scalable, maintainable, and performant.

The system can receive a lot of future work because of its modularity and scalability. Even though much has been done to enhance performance, there may still be room for improvement. Furthermore, the addition of other programming languages can significantly improve the system. Finally, providing more public tests and the ability to detect more design patterns for the languages that are already supported will be a simple and quick but very rewarding upgrade.

REFERENCES

[1] S. K. Basak, M. Wotto, and P. Bélanger, “E-learning, M-learning and D-learning: Conceptual definition and comparative analysis,” *E-Learning and Digital Media*, vol. 15, no. 4, pp. 191–216, 2018, doi: 10.1177/2042753018785180.

[2] H. L. Steen, “Effective eLearning design,” *MERLOT Journal of Online Learning and Teaching*, vol. 4, no. 4, pp. 526–532, 2008.

[3] LeetCode, “The World’s Leading Online Programming Learning Platform.” https://leetcode.com/ (accessed Apr. 09, 2022).

[4] Hackerrank, “Hackerrank.” https://www.hackerrank.com/ (accessed Apr. 09, 2022).

[5] CodinGame, “Coding Games and Programming Challenges to Code Better.” https://www.codingame.com/start (accessed Apr. 09, 2022).

[6] Codewars, “Achieve Mastery through Coding Challenge.” https://www.codewars.com/ (accessed Apr. 09, 2022).

[7] Coderbyte, “Improve Your Coding Skills.” https://www.coderbyte.com/. (accessed Apr. 09, 2022).

[8] Topcoder, “Top Website Designers, Developers, Freelancers for Your next Project.” https://www.topcoder.com/ (accessed Apr. 09, 2022).

[9] E. G. Dada, A. H. Alkali, and D. O. Oyewola, “An investigation into the effectiveness of asynchronous and synchronous e-learning mode on students’ academic performance in National Open University (NOUN), Maiduguri Centre,” *International Journal of Modern Education and Computer Science*, vol. 10, no. 5, p. 54, 2019.

[10] B. Aguti, “A Model to facilitate effective E-learning in technology-enhanced learning environments within universities,” University of Southampton, 2015.

[11] E. H.-K. Wu, C.-H. Lin, Y.-Y. Ou, C.-Z. Liu, W.-K. Wang, and C.-Y. Chao, “Advantages and constraints of a hybrid model K-12 E-Learning assistant chatbot,” *Ieee Access*, vol. 8, pp. 77788–77801, 2020.

[12] F. Amiti, “Synchronous and asynchronous E-learning,” *European Journal of Open Education and E-Learning Studies*, vol. 5, no. 2, 2020.

[13] I. Yuhanna, A. Alexander, and A. Kachik, “Advantages and disadvantages of Online Learning,” *Journal Educational Verkenning*, vol. 1, no. 2, pp. 13–19, 2020.

[14] I.-C. Tudorache, A.-M. M. Iordache, and M.-T. Iordache, “The advantages and the disadvantages of e-learning in European union,” *Journal of Information Systems & Operations Management*, vol. 6, no. 2, p. 1, 2012.

[15] V. Arkorful and N. Abaidoo, “The role of e-learning, advantages and disadvantages of its adoption in higher education,” *International journal of instructional technology and distance learning*, vol. 12, no. 1, pp. 29–42, 2015.

[16] M. Tuteja, G. Dubey, and others, “A research study on importance of testing and quality assurance in software development life cycle (SDLC) models,” *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 3, pp. 251–257, 2012.

[17] I. Sheriguda and R. Reddy, “Studying the Importance and Feasibility of Software Testing”.

[18] T. Jindal, “Importance of Testing in SDLC,” *International Journal of Engineering and Applied Computer Science (IJEACS)*, vol. 1, no. 02, pp. 54–56, 2016.

[19] P. S. Kochhar, X. Xia, and D. Lo, “Practitioners’ views on good software testing practices,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019, pp. 61–70.

[20] B. George and L. Williams, “A structured experiment of test-driven development,” *Information and software Technology*, vol. 46, no. 5, pp. 337–342, 2004.

[21] D. Janzen and H. Saiedian, “Test-driven development concepts, taxonomy, and future direction,” *Computer (Long Beach Calif)*, vol. 38, no. 9, pp. 43–50, 2005.

[22] D. Janzen and H. Saiedian, “Does test-driven development really improve software design quality?,” *Ieee Software*, vol. 25, no. 2, pp. 77–84, 2008.

[23] M. Olan, “Unit testing: test early, test often,” *Journal of Computing Sciences in Colleges*, vol. 19, no. 2, pp. 319–328, 2003.

[24] P. Runeson, “A survey of unit testing practices,” *IEEE Softw*, vol. 23, no. 4, pp. 22–29, 2006.

[25] F. Trautsch, S. Herbold, and J. Grabowski, “Are unit and integration test definitions still valid for modern Java projects? An empirical study on open-source projects,” *Journal of Systems and Software*, vol. 159, p. 110421, 2020.

[26] Z. Jin and A. J. Offutt, “Coupling-based criteria for integration testing,” *Software Testing, Verification and Reliability*, vol. 8, no. 3, pp. 133–154, 1998.

[27] H. K. N. Leung and L. White, “A study of integration testing and software regression at the integration level,” in *Proceedings. Conference on Software Maintenance 1990*, 1990, pp. 290–301.

[28] A. Laapas and others, “Cost-Benefit analysis of using test automation in the development of embedded software,” 2014.

[29] M. M. Moe, “Comparative Study of Test-Driven Development TDD, Behavior-Driven Development BDD and Acceptance Test–Driven Development ATDD,” *International Journal of Trend in Scientific Research and Development*, vol. 3, pp. 231–234, 2019.

[30] E. Collins, A. Dias-Neto, and V. F. de Lucena Jr, “Strategies for agile software testing automation: An industrial experience,” in *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, 2012, pp. 440–445.

[31] R. C. Martin, “Design principles and design patterns,” *Object Mentor*, vol. 1, no. 34, p. 597, 2000.

[32] C. Y. Baldwin and K. B. Clark, “The architecture of participation: Does code architecture mitigate free riding in the open source development model?,” *Manage Sci*, vol. 52, no. 7, pp. 1116–1127, 2006.

[33] D. Soni, R. L. Nord, and C. Hofmeister, “Software architecture in industrial applications,” in *1995 17th International Conference on Software Engineering*, 1995, p. 196.

[34] H. M. dos Santos, V. H. S. Durelli, M. Souza, E. Figueiredo, L. T. da Silva, and R. S. Durelli, “Cleangame: Gamifying the identification of code smells,” in *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, 2019, pp. 437–446.

[35] M. v Mantyla, J. Vanhanen, and C. Lassenius, “Bad smells-humans as code critics,” in *20th IEEE International Conference on Software Maintenance, 2004. Proceedings.*, 2004, pp. 399–408.

[36] M. Mantyla, J. Vanhanen, and C. Lassenius, “A taxonomy and an initial empirical study of bad smells in code,” in *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.*, 2003, pp. 381–384.

[37] M. Fowler, *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.

[38] A. Yamashita and L. Moonen, “To what extent can maintenance problems be predicted by code smell detection?–An empirical study,” *Information and Software Technology*, vol. 55, no. 12, pp. 2223–2242, 2013.

[39] D. I. K. Sjøberg, A. Yamashita, B. C. D. Anda, A. Mockus, and T. Dybå, “Quantifying the effect of code smells on maintenance effort,” *IEEE Transactions on Software Engineering*, vol. 39, no. 8, pp. 1144–1156, 2012.

[40] E. Gamma, R. Helm, R. Johnson, R. E. Johnson, J. Vlissides, and others, *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH, 1995.

[41] B. Turkistani and Y. Liu, “Reducing the large class code smell by applying design patterns,” in *2019 IEEE International Conference on Electro Information Technology (EIT)*, 2019, pp. 590–595. Accessed: Apr. 09, 2022. [Online]. Available: https://www.topcoder.com/.

[42] T. Alkhaeir and B. Walter, “The effect of code smells on the relationship between design patterns and defects,” *IEEE Access*, vol. 9, pp. 3360–3373, 2020.

[43] B. Walter and T. Alkhaeir, “The relationship between design patterns and code smells: An exploratory study,” *Information and Software Technology*, vol. 74, pp. 127–142, 2016, doi: https://doi.org/10.1016/j.infsof.2016.02.003.

[44] M. Ali and M. O. Elish, “A Comparative Literature Survey of Design Patterns Impact on Software Quality,” in *2013 International Conference on Information Science and Applications (ICISA)*, 2013, pp. 1–7. doi: 10.1109/ICISA.2013.6579460.

[45] C. Zhang and D. Budgen, “What Do We Know about the Effectiveness of Software Design Patterns?,” *IEEE Transactions on Software Engineering*, vol. 38, no. 5, pp. 1213–1231, 2012, doi: 10.1109/TSE.2011.79.

[46] D. Alur, J. Crupi, and D. Malks, *Core J2EE patterns: best practices and design strategies*. Gulf Professional Publishing, 2003.

[47] M. Fowler, *Patterns of Enterprise Application Architecture: Pattern Enterpr Applica Arch*. Addison-Wesley, 2012.

[48] P. Jamshidi, C. Pahl, N. C. Mendonça, J. Lewis, and S. Tilkov, “Microservices: The journey so far and challenges ahead,” *IEEE Software*, vol. 35, no. 3, pp. 24–35, 2018.

[49] S. and L. A. L. and M. M. and M. F. and M. R. and S. L. Dragoni Nicola and Giallorenzo, “Microservices: Yesterday, Today, and Tomorrow,” in *Present and Ulterior Software Engineering*, B. Mazzara Manuel and Meyer, Ed. Cham: Springer International Publishing, 2017, pp. 195–216. doi: 10.1007/978-3-319-67425-4_12.

[50] C. Pahl and P. Jamshidi, “Microservices: A Systematic Mapping Study.,” *CLOSER (1)*, pp. 137–146, 2016.

[51] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, “Microservices,” *IEEE Software*, vol. 35, no. 3, pp. 96–100, 2018.

[52] Martin Fowler and James Lewis, “monolithic architecture vs microservices architecture,” Mar. 25, 2014. https://martinfowler.com/articles/microservices/images/decentralised-data.png (accessed Apr. 06, 2022).

[53] J. Ghofrani and D. Lübke, “Challenges of Microservices Architecture: A Survey on the State of the Practice.,” *ZEUS*, vol. 2018, pp. 1–8, 2018.

[54] Y. Wang, H. Kadiyala, and J. Rubin, “Promises and challenges of microservices: an exploratory study,” *Empirical Software Engineering*, vol. 26, no. 4, pp. 1–44, 2021.

[55] J. Fritzsch, J. Bogner, S. Wagner, and A. Zimmermann, “Microservices migration in industry: intentions, strategies, and challenges,” in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2019, pp. 481–490.

[56] C. Esposito, A. Castiglione, and K.-K. R. Choo, “Challenges in delivering software in the cloud as microservices,” *IEEE Cloud Computing*, vol. 3, no. 5, pp. 10–14, 2016.

[57] D. Taibi, V. Lenarduzzi, and C. Pahl, “Continuous architecting with microservices and devops: A systematic mapping study,” in *International Conference on Cloud Computing and Services Science*, 2018, pp. 126–151.

[58] D. Taibi, V. Lenarduzzi, C. Pahl, and A. Janes, “Microservices in agile software development: a workshop-based study into issues, advantages, and disadvantages,” in *Proceedings of the XP2017 Scientific Workshops*, 2017, pp. 1–5.

[59] D. Taibi, V. Lenarduzzi, and C. Pahl, “Architectural patterns for microservices: a systematic mapping study,” 2018.

[60] K. Li, J. Xiao, Y. Wang, and Q. Wang, “Analysis of the key factors for software quality in crowdsourcing development: An empirical study on topcoder. com,” in *2013 IEEE 37th Annual Computer Software and Applications Conference*, 2013, pp. 812–817.

[61] M. V. P. de Almeida, “EasyCoding-methodology to support programming learning,” 2021.

[62] V. Spichak and S. Petrov, “Experience in Designing and Developing the Educational Game BlockSolver,” in *2020 V International Conference on Information Technologies in Engineering Education (Inforino)*, 2020, pp. 1–5.

[63] K. Buzzard, “Proving theorems with computers,” *Notices of the American Mathematical Society*, vol. 67, no. 11, p. 1, 2020.

[64] J. I. F. Rosado, “Video Games to learn programming,” *Revista Educación en Ingenier\'\ia*, vol. 14, no. 28, pp. 119–123, 2019.

[65] S. Combéfis, G. Beresnevičius, and V. Dagienė, “Learning programming through games and contests: overview, characterisation and discussion,” *Olympiads in Informatics*, vol. 10, no. 1, pp. 39–60, 2016.

[66] T. di Mascio, L. Laura, and M. Temperini, “A framework for personalized competitive programming training,” in *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2018, pp. 1–8.

[67] I. S. Zinovieva *et al.*, “The use of online coding platforms as additional distance tools in programming education,” in *Journal of Physics: Conference Series*, 2021, vol. 1840, no. 1, p. 12029.

[68] D. Heller, *Building a Career in Software*. Springer, 2020.

[69] T. Camp *et al.*, “Generation CS: the growth of computer science,” *ACM Inroads*, vol. 8, no. 2, pp. 44–50, 2017.

[70] C. L. Gordon, R. Lysecky, and F. Vahid, “The rise of program auto-grading in introductory cs courses: A case study of zylabs,” 2021.

[71] F. Restrepo-Calle, J. J. Ram\'\irez Echeverry, and F. A. González, “Continuous assessment in a computer programming course supported by a software tool,” *Computer Applications in Engineering Education*, vol. 27, no. 1, pp. 80–89, 2019.

[72] B. B. Rad, H. J. Bhatti, and M. Ahmadi, “An introduction to docker and analysis of its performance,” *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 3, p. 228, 2017.

[73] C. Anderson, “Docker [software engineering],” *Ieee Software*, vol. 32, no. 3, pp. 102–c3, 2015.

[74] IBM Cloud Education, “Docker,” Jun. 23, 2021. https://www.ibm.com/cloud/learn/docker (accessed Jun. 16, 2022).

[75] Docker, “Docker overview.” https://docs.docker.com/get-started/overview/ (accessed Jun. 16, 2022).

[76] SIMRAN ARORA, “Docker vs. Virtual Machines,” Oct. 29, 2019. https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/ (accessed Jun. 16, 2022).

[77] Atul Kumar, “Docker Architecture,” Apr. 26, 2021. https://k21academy.com/docker-kubernetes/docker-tutorial/ (accessed Jun. 16, 2022).

[78] V. Medel, R. Tolosana-Calasanz, J. Á. Bañares, U. Arronategui, and O. F. Rana, “Characterising resource management performance in Kubernetes,” *Computers & Electrical Engineering*, vol. 68, pp. 286–297, 2018.

[79] V. Medel, O. Rana, J. Á. Bañares, and U. Arronategui, “Modelling performance & resource management in kubernetes,” in *Proceedings of the 9th International Conference on Utility and Cloud Computing*, 2016, pp. 257–262.

[80] Docker, “Kubernetes.” https://www.docker.com/products/kubernetes/ (accessed Jun. 16, 2022).

[81] Kubernetes, “Kubernetes.” https://kubernetes.io/ (accessed Jun. 16, 2022).

[82] Red Hat, “What is Kubernetes?,” Mar. 27, 2020. https://www.redhat.com/en/topics/containers/what-is-kubernetes (accessed Jun. 16, 2022).

[83] Facebook, “React.” https://reactjs.org/ (accessed Jun. 16, 2022).

[84] S. Aggarwal, “Modern web-development using reactjs,” *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 133–137, 2018.

[85] S. Aggarwal and J. Verma, “Comparative analysis of MEAN stack and MERN stack,” *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 127–132, 2018.

[86] Facebook, “JSX.” https://facebook.github.io/jsx/#sec-intro (accessed Jun. 16, 2022).

[87] Mozilla, “What is JavaScript?” https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (accessed Jun. 16, 2022).

[88] D. Crockford, *JavaScript: The Good Parts: The Good Parts*. “ O’Reilly Media, Inc.,” 2008.

[89] P. Wilton, *Beginning JavaScript*. John Wiley & Sons, 2004.

[90] A. Feldthaus and A. Møller, “Checking correctness of TypeScript interfaces for JavaScript libraries,” in *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications*, 2014, pp. 1–16.

[91] S. Fenton, Fenton, and Spearing, *Pro TypeScript*. Springer, 2014.

[92] S. Tilkov and S. Vinoski, “Node. js: Using JavaScript to build high-performance network programs,” *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.

[93] N. Chhetri, “A comparative analysis of node. js (server-side javascript),” 2016.

[94] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, “A comparative study: MongoDB vs. MySQL,” in *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, 2015, pp. 1–6.

[95] MongoDB, “Build faster. Build smarter..” https://www.mongodb.com/ (accessed Jun. 16, 2022).

[96] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.

[97] Express, “Fast, unopinionated, minimalist web framework for Node.js.” https://expressjs.com/ (accessed Jun. 16, 2022).

[98] A. Mardan, *Express. js Guide: The Comprehensive Book on Express. js*. Azat Mardan, 2014.

[99] E. Hahn, *Express in Action: Writing, building, and testing Node. js applications*. Simon and Schuster, 2016.

[100] J. N. Robbins, *Learning web design: A beginner’s guide to HTML, CSS, JavaScript, and web graphics*. “ O’Reilly Media, Inc.,” 2012.

[101] HTML, “HTML For Beginners The Easy Way: Start Learning HTML & CSS Today.” https://html.com/ (accessed Jun. 16, 2022).

[102] Mozilla, “CSS: Cascading Style Sheets”, Accessed: Jun. 16, 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS