Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Sultan Saliev Azamat Rsymbetov

algomalgo.com - Web Platform for learning Algorithms

A thesis submitted for the degree of
Bachelor in Computer Science
(degree code: 6B06102)

Kaskelen, 2020

Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University
Faculty of Engineering and Natural Sciences

algomalgo.com - Web Platform for learning Algorithms

A thesis submitted for the degree of
Bachelor in Computer Science
(degree code: 6B06102)

Authors: **Sultan Saliev Azamat Rsymbetov**

Supervisor: **Ualikhan Sadyk**

Dean of the faculty:
Assist. Prof. Andrey Bogdanchikov

Kaskelen, 2020

Abstract

This thesis discusses the gamification or use of game elements in non-game contexts, as well as its potential use in education to enhance student motivation and involvement. The use of technology helps to increase the effectiveness of training. Mobile devices are designed to be powerful mini-computers, and the potential for learning on these devices is growing. This means that people can learn using interactive technology whenever and wherever they wish using their own personal devices. You can add different elements of gamification to different types of training. In the future, it will be possible to integrate this method into state institutions, such as schools and universities.

Аңдатпа

Диссертациялық жұмыста ойын элементтерін ойынға қосу немесе қолдану, сондай-ақ оқушылардың ынтасы мен белсенділігін арттыру үшін оны білім беруде қолдану мүмкіндігі қарастырылады. Технологияны қолдану оқытудың тиімділігін арттыруға көмектеседі. Мобильді құрылғылар қуатты мини-компьютерлерге арналған, сондықтан бұл құрылғыларда білім алу мүмкіндігі артып келеді. Бұл дегеніміз, адамдар интерактивті технологияны пайдалана отырып, жеке құрылғыларды қашан және қай жерде қолданғысы келетіндігін біле алады. Әр түрлі жаттығуларға геймификацияның әр түрлі элементтерін қосуға болады. Болашақта бұл әдісті мемлекеттік мекемелерге, мысалы мектептер мен университеттерге біріктіруге болады.

Аннотация

В этом тезисе обсуждается геймификация или использование игровых элементов в неигровых контекстах, а также ее потенциальное применение в образовании для усиления мотивации и вовлеченности учащихся. Использование технологий способствует повышению эффективности обучения. Мобильные устройства разрабатываются для того, чтобы они были мощными миникомпьютерами, и потенциал для обучения на этих устройствах становится все больше. Это означает, что люди могут учиться, используя интерактивные технологии, когда и где они пожелают, используя свои собственные персональные устройства. Можно добавить различные элементы геймификации в различные типы обучения. В дальнейшем можно будет интегрирывать этот метод в государственные учереждения, такие как школы и вузы.

Contents

Chapter 1

Introduction

1.1 Motivation

Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output. We can also view an algorithm as a tool for solving a well-specified computational problem. The statement of the problem specifies in general terms the desired input/output relationship. The algorithm describes a specific computational procedure for achieving that input/output relationship.

For example, we might need to sort a sequence of numbers into nondecreasing order. This problem arises frequently in practice and provides fertile ground for introducing many standard design techniques and analysis tools. Here is how we formally define the sorting problem:

Input: $\langle a_1, a_2, a_3...a_n \rangle$
Output: A permutation (reordering) $\langle a_1^{'}, a_2^{'}, a_3^{'}...a_n^{'} \rangle$

For example, given the input sequence $\langle 31, 41, 59, 26, 41, 58 \rangle$ a sorting algorithm returns as output the sequence $\langle 26, 31, 41, 41, 58, 59 \rangle$. Such an input sequence is called an instance of the sorting problem. In general, an instance of a problem consists of the input (satisfying whatever constraints are imposed in the problem statement) needed to compute a solution to the problem.

In the present age, people who aspire to work in those fields learn frameworks and libraries that abstract away the use of algorithms and hide the computational cost of each used function. Generally, that is a good thing, because it greatly improves their workflow and makes those professions more accessible, thus reducing the costs of production. However, the downsides are that it affects the software quality, decreasing performance and using large amounts of memory. One of the recent examples of algorithm misuse that appeared globally is a GTA V game, where the if statement was triggered 1.98 billion times while preloading the GTA V online screen.

As you can see, algorithms are the basis of all computational problems, thus it could be seen as an essential tool for everyone in the computer science field including: software engineers, computer science researchers, data scientists, machine learning engineers, data engineers, etc.

1.2 Aims and Objectives

Our project is aimed to make a decent contribution to educating people who aspire to enter career in Computer Science related professions.

1.3 Thesis Outline

The first chapter is an *Introduction* chapter. This chapter introduces you to gamification and describes aims of this thesis. The second chapter is **??**, which describes the gaming industry. Since it is important to find out why people like to play games, and how to direct this interest in the right direction. Chapter three is the *The Death of Moore's Law*, which explains about implementation experience and basic gamification techniques. In chapter *Pre-existing solutions* observations and comparisons of gamification material are described.

Chapter 2

The Origin of Algorithms

2.1 Matematician from Uzbekistan

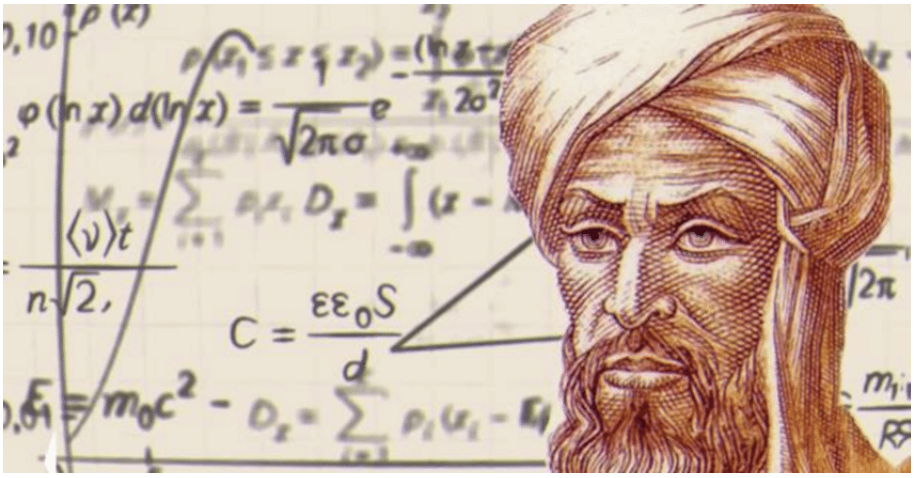


Figure 2.1: Abu Abdullah Muhammad ibn Musa Al-Khawarizmi

The term algorithm got its name from the Persian astronomer and mathematician, Abu Abdullah Muhammad ibn Musa Al-Khawarizmi (780 AD).
He was from a Persian city known as Khwarizm, found in present-day Uzbekistan.
Al-Khwarizmi wrote many important mathematical books that were later translated into Latin.
It was his book on Arabic-Hindu numerals and arithmetic, called Al-Khw?rizm? On the Hindu Art of Reckoning, later Latinized as Algoritmi de numero Indorum that gave the word algorithm to the Western world.

2.2 First Algorithm

Figure 2.2: Homo Erectus

Generally, the existence of algorithms as long as human history. As an example, the creation of the first artificial fire in the Wonderwerk Cave of South Africa, millions of years ago by Homo Erectus, constitutes our first evidence of the use of algorithms. However, most historians hold the view that the Babylonian clay tablets (1600-1800 BC) are the world's first known algorithm. The Babylonians had developed a numerical system using cuneiform numerals to count and they preserved those calculations on tablets.

Figure 2.3: Cuneiform tablet

Although we can find instances of algorithms in Euclidean mathematics, Archimedes approximation of Pi, or Eratosthenes calculation of prime numbers, it was the work of George Boole in developing what is known as Boolean algebra, in 1847, that set the basis for computer logic today. Boole developed a system of logic to make calculations that used true and false values as basic units (binary algebra). This system would later be represented in digital form by zeros and ones or binary digits in low-level programming languages.

2.3 Computing machine

Some decades later, Alan Turing came up with a mathematical model of a hypothetical computing machine. It used squares that contained symbols or binary digits in an infinitely long tape. This tape served as storage space for the data in the squares, computer memory. It has a head or needle that can move to the right or left of each square to read, write, or erase the symbol within it, the basis for CRUD (Create, Read, Update, and Delete) operations in programming. Using this system this machine could simulate any algorithm, regardless of its complexity! Mathematicians and computer scientists continued adding to Turing's concept of a computing machine to advance technology to what we know today. Algorithms then evolved from binary operations to high-level, more human-friendly, programming languages like C++, Java, and etc. Because they allow creating efficient, error-free software applications, they will continue to be important in the foreseeable future.

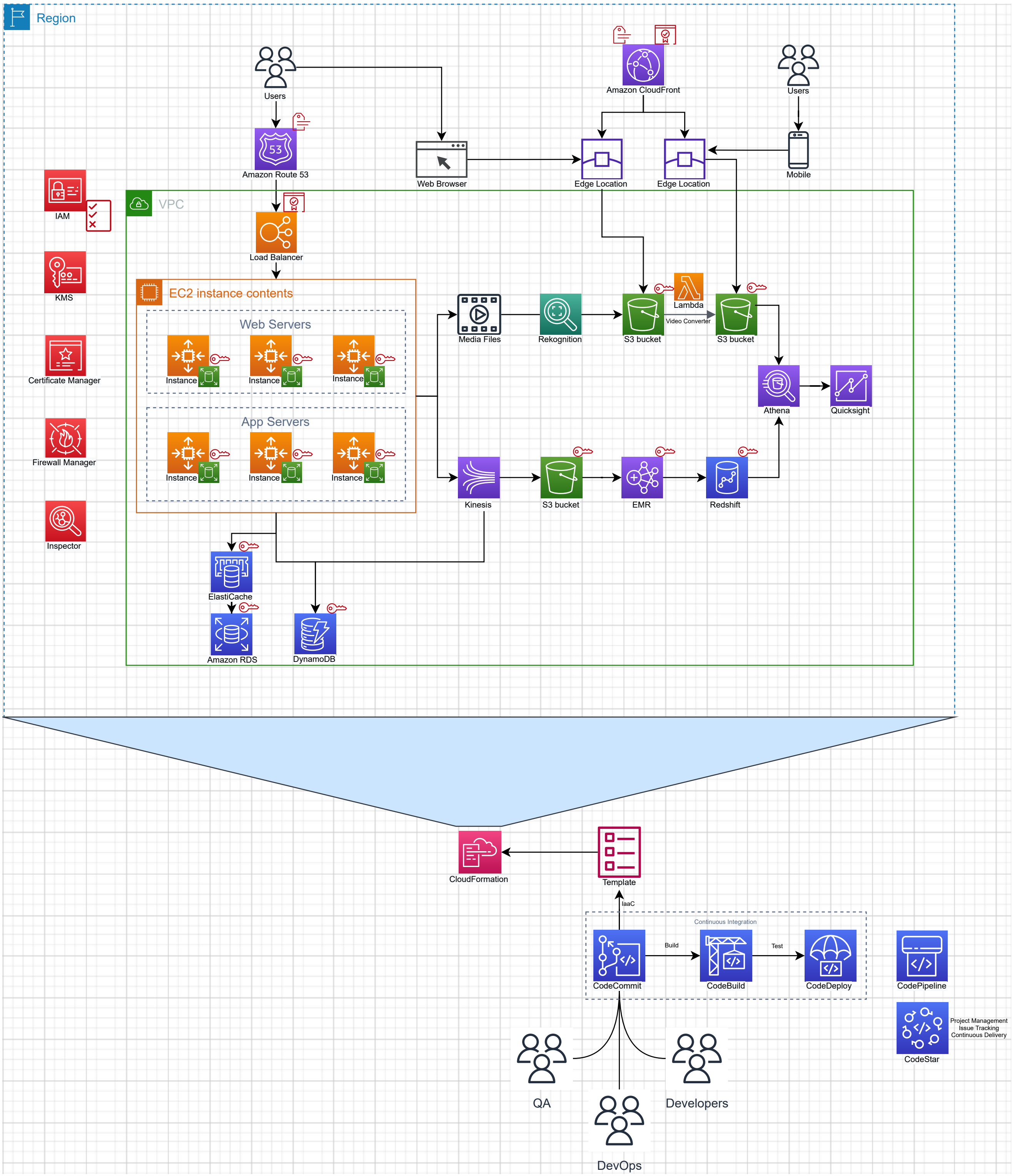


Figure 2.4: System Design

Chapter 3

The Death of Moore's Law

3.1 Introduction

The advent of RISC computing in the 1980s was the basis for Gordon Moore's initial prediction and faithfully showed a doubling in processor performance every 18 months. But, as the limits of clock frequency per chip began to appear, the use of Dennard scaling and multicore CPUs helped prolong the performance curve. But it is important to note that even at the start of the century, we were no longer on the Moore's Law curve, and doubling of performance took 3.5 years during this time.

Amdahl's Law refers to the limits of performance improvement that can be achieved with parallel processing. While parallelizing the execution of a process can provide an initial performance boost, there will always be a natural limit, as there are some execution tasks that cannot be parallelized. We have recently experienced that these limits come into effect when the benefits of using multiple CPU cores decrease, leading to an even longer time span between performance improvements.

The prediction, as can be seen in the graph, is that it will now take 20 years for CPU processing power to double in performance. Hence, Moore's Law is dead.

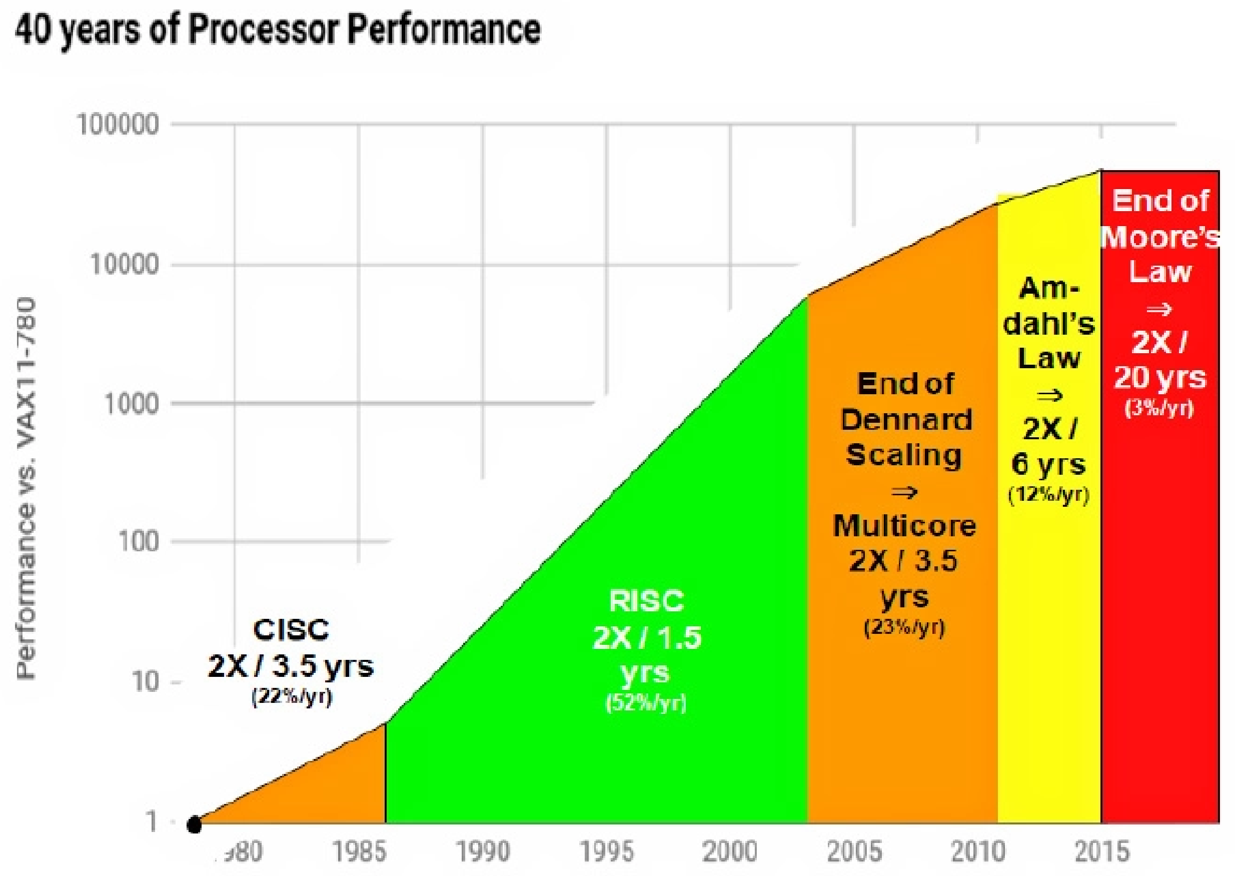


Figure 3.1: Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

However, the underlying business model assumption here is that as the number of clients and volume of work increases, it is enough to simply add more servers. But, as can be clearly seen in the earlier graph, server processing performance will only grow 3% per year over the next 20 years. This is far below the expectation that the amount of data to be processed will triple over the next five years That's why the efficient use of that hardware is more important than ever and that requires the knowledge of algorithms and data structures.

3.2 Algorithms in Big Tech Companies

Most of the companies as a basis of their success used advanced algorithms to scale their service. One of the popular examples is Google, which used advanced web crawler technology and web site indexing to find information across the web. Google's web crawler algorithm works in such a way, where it explores new sites existing on the web, based on visiting pre-existing site's links, analyzing over 40,000 new pages in the web per second. After a page is crawled, Google tries to understand what the page is about. This stage is called indexing and it includes processing and analyzing the textual content and key content tags and attributes, such as <title> elements and alt attributes, images, videos, and more. All the process could be seen as a graph traversal procedure, where its page could be seen as a single node and links between pages could be viewed as edges. Using these technologies, they became a one of the significal companies in their field.

Another decent example could be Alibaba group, which used advanced algorithms in their cloud computing service in order to perform well on Single's day, when the number of orders were at its peak. During this day Alibaba cloud processed 175,000 transactions and 120,000 per second at peak traffic, earning for a company around 84.54 billion U.S. dollars for two weeks according to 2021 yearly report. In order to perform well, they have used the Apache Flink framework, to distribute real time data across many clusters. However, to divide that data into meaningful chunks, they have used a sliding window algorithm, which is one of the most popular algorithms. Thus, the knowledge of good algorithms helped them to achieve their goal.

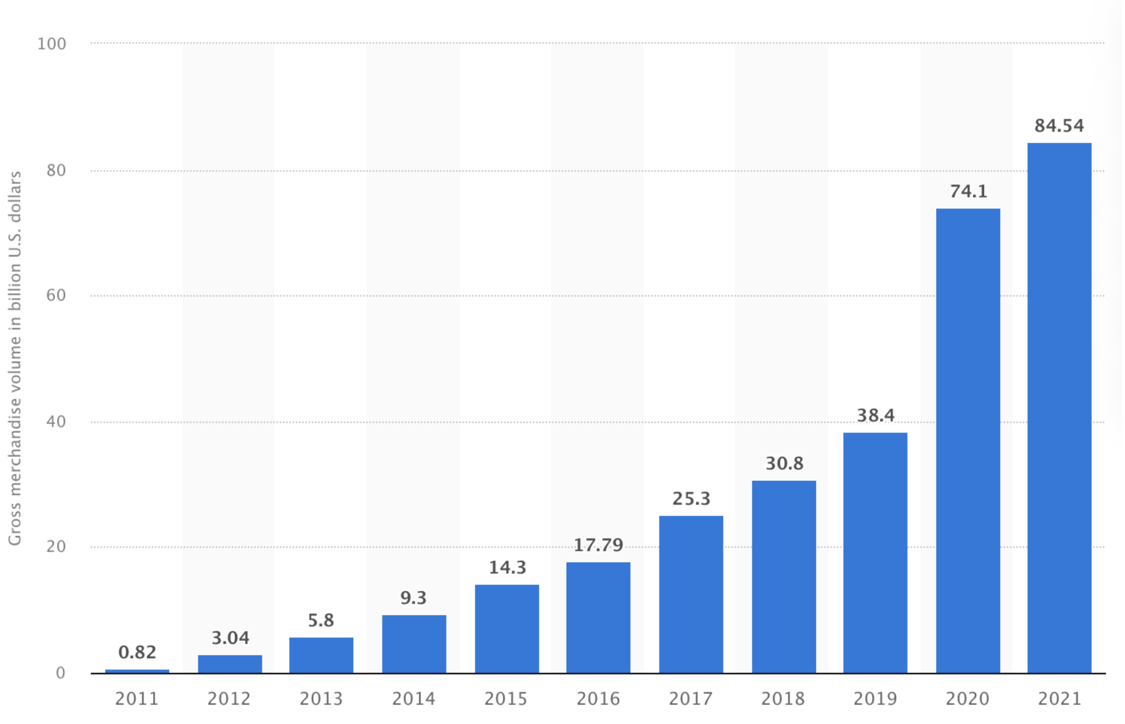


Figure 3.2: Alibaba's gross merchandise volume on Singles' Day from 2011 to 2021 (in billion U.S. dollars)

The other example could be Amazon, which incrementally updated the quality of their services in order to perform well on the market. At first, they only had e-commerce website with some goods, however to increase interest rate, they started to invent various algorithms, one of the examples is collaborative filtering algorithm, which was revolutionary 20 years ago. After a few years they have improved their algorithm adding some heuristics and by increasing computational power they made items on their site more customizable for a specific person, which greatly boosted the amount of their revenue in Amazon, Amazon Kindle and Amazon Prime. As for today, they use multilayered collaborative neural networks, to make the best possible predictions, which will be beneficial for their business.

Uber could be another good example, they use sophisticated path finding algorithms as Dijkstra and A star for cost and time optimization purposes. Using these algorithms at their core, they have created a good customer experience, which led to its becoming a multi-billion company.

Chapter 4

Pre-existing solutions

4.1 Introduction

It appears that there are several solutions, where people could find decent information for studying algorithms for their purposes. Goals for studying algorithms could vary from preparing for a big tech company's technical interview to passing an assignment or understanding general principles behind popular frameworks. Discouragingly, there are too few good services that could be helpful for one's aims. Popular solutions such as Video lectures or textbooks could contain a lot of unnecessary information and can be time consuming on its own. In addition to that they could contain misinformation. As an example Binary Search algorithm, which has been described in several textbooks contained wrong implementation causing Integer Overflow problem. Generally, for finding middle element it was common to add high and low elements dividing by two. However, right implementation for finding middle element would be subtracting high element to low element, dividing the product.

4.2 Market research

Other solutions like: AlgoExpert, Brilliant, GeekForGeeks or CP-algorithms may contain vague information or could be too focused for a specific group of users. Let's tackle them one by one:

4.2.1 Brilliant

Brilliant focuses on general STEM knowledge and has a variety of topics, but it lacks in depth and focus. It is good for learning the general idea about STEM subjects and to improve your erudition. In addition to that it has good visuals and interactive content mostly consisting of small puzzles. However, it is not a place to learn and deeply understand algorithms and data structure, which could be implemented in day to day work.

4.2.2 GeeksForGeeks

GeeksForGeeks is very similar to W3Schools.com with the focus on general Computer Science topics. However, it lacks good structure and organization of topics. In addition to that, their website's UI and UX makes it more difficult to navigate and find the content you need. In most cases, people find topics in GeeksForGeeks through search engines such as Google, instead of using GeeksForGeeks. It is good for finding general information about a specific topic and quickly learning it.

4.2.3 AlgoExpert

AlgoExpert focused more on interview preparation. It has a built in coding environment and a video explanation for each coding problem. Their coding problems are hand-picked for coding interview preparation only. It is good for preparing for coding interviews in a short amount of time.

4.2.4 CP-algorithms

CP-algorithms is similar to GeeksForGeeks, but the content is only focused on algorithms related to competitive programming, but the explanations of the topics are more deep and advanced, which makes it definitely not beginner friendly.

4.3 Ultimate Solution

The purpose of this project is to provide the solution to learn algorithms in a structured way with proper depth and free of charge. It won't be as surface level as brilliant but won't go too deep like cp-algorithms. Our project is also more foundational and not too focused on specific things like AlgoExpert. It also will be easier to navigate compared to GeeksForGeeks and you will be able to track your progress.

So to conclude, our project combines the good sides of our competitors and provides a more well-rounded solution.

Chapter 5

Implementation

5.0.1 Tech Stack

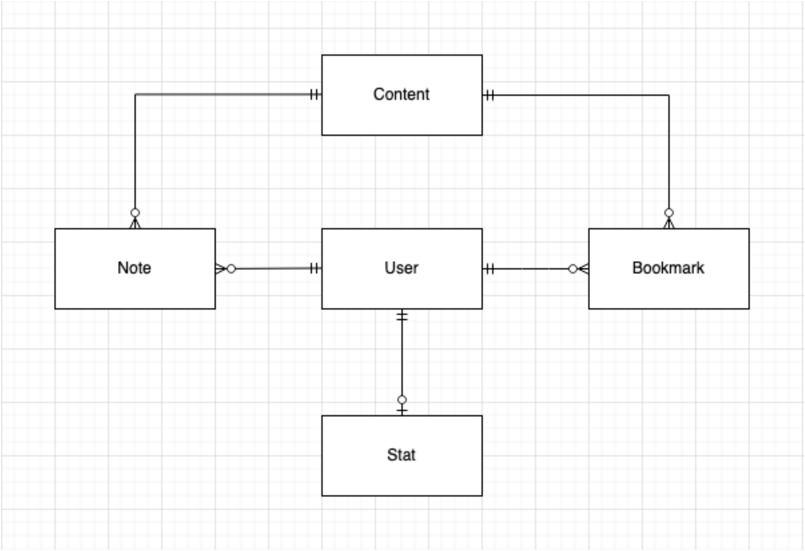


Figure 5.1: Entity Relationship Diagram

Frontend: React.js, Tailwind.css, Next.js
Backend: Next.js
Database: PostgreSQL

5.0.2 Wireframes

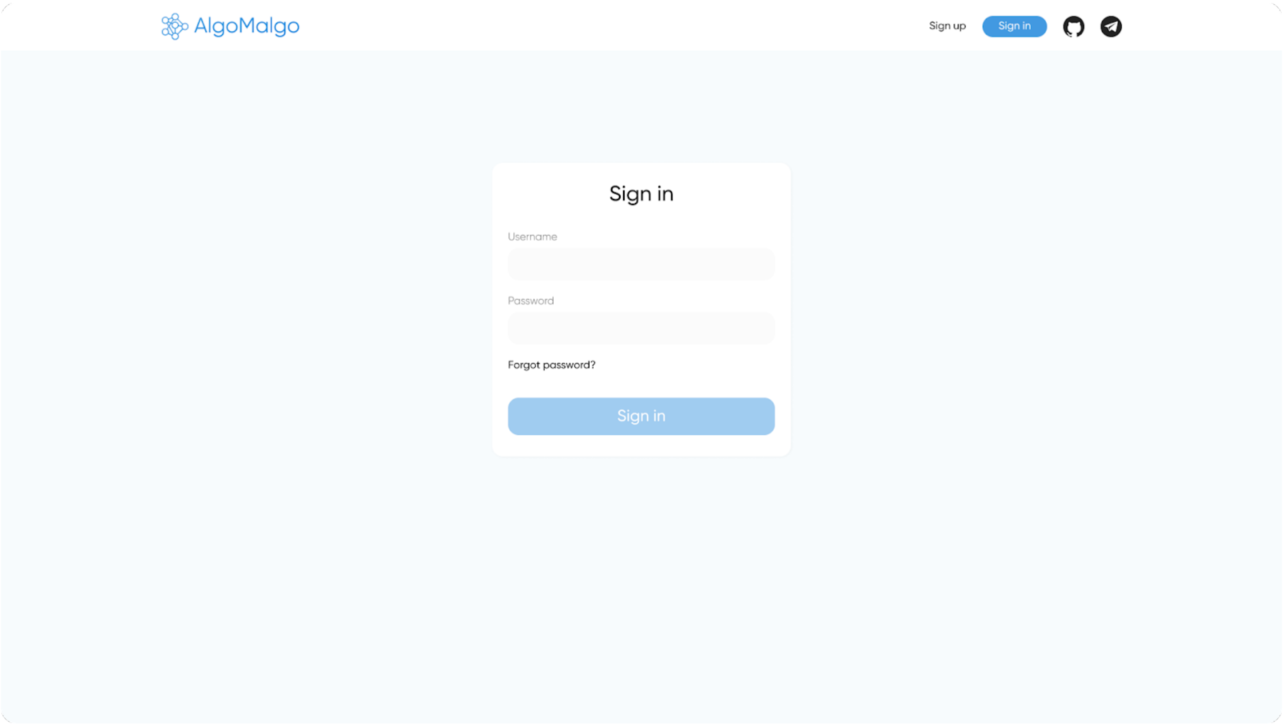
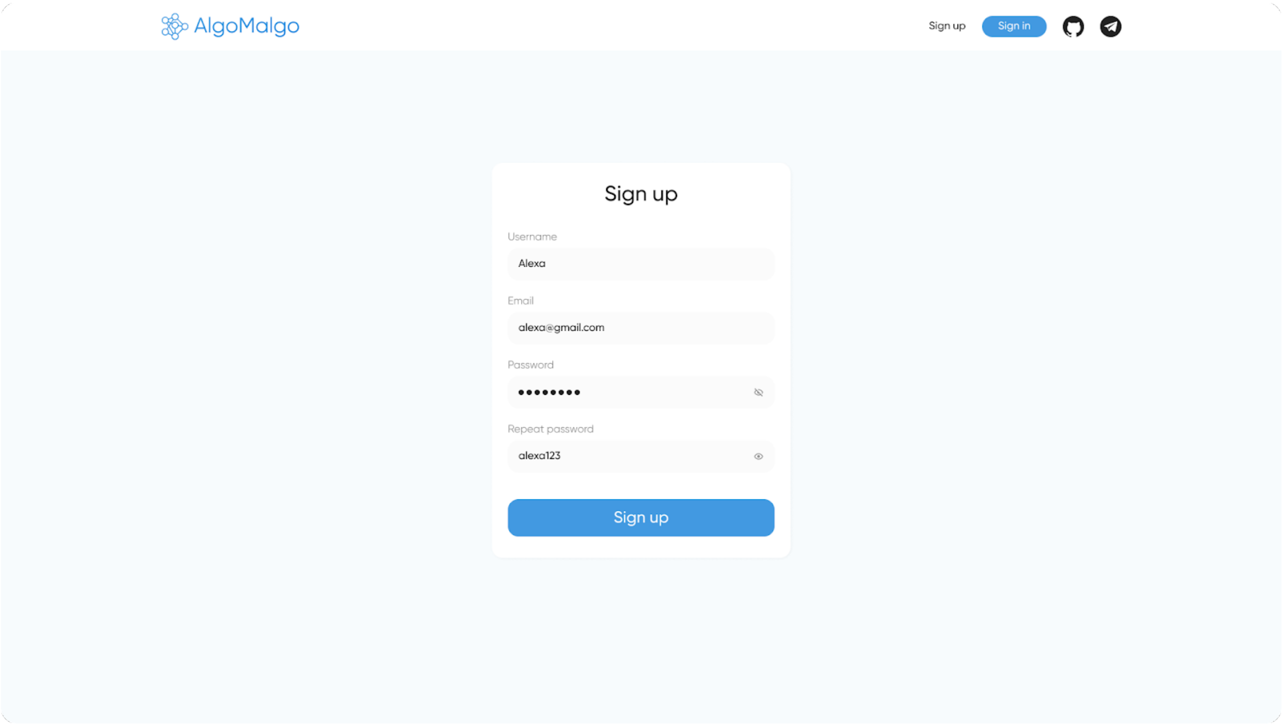


Figure 5.2: Sign In



Username

Alexa

Email

alexa@gmail.com

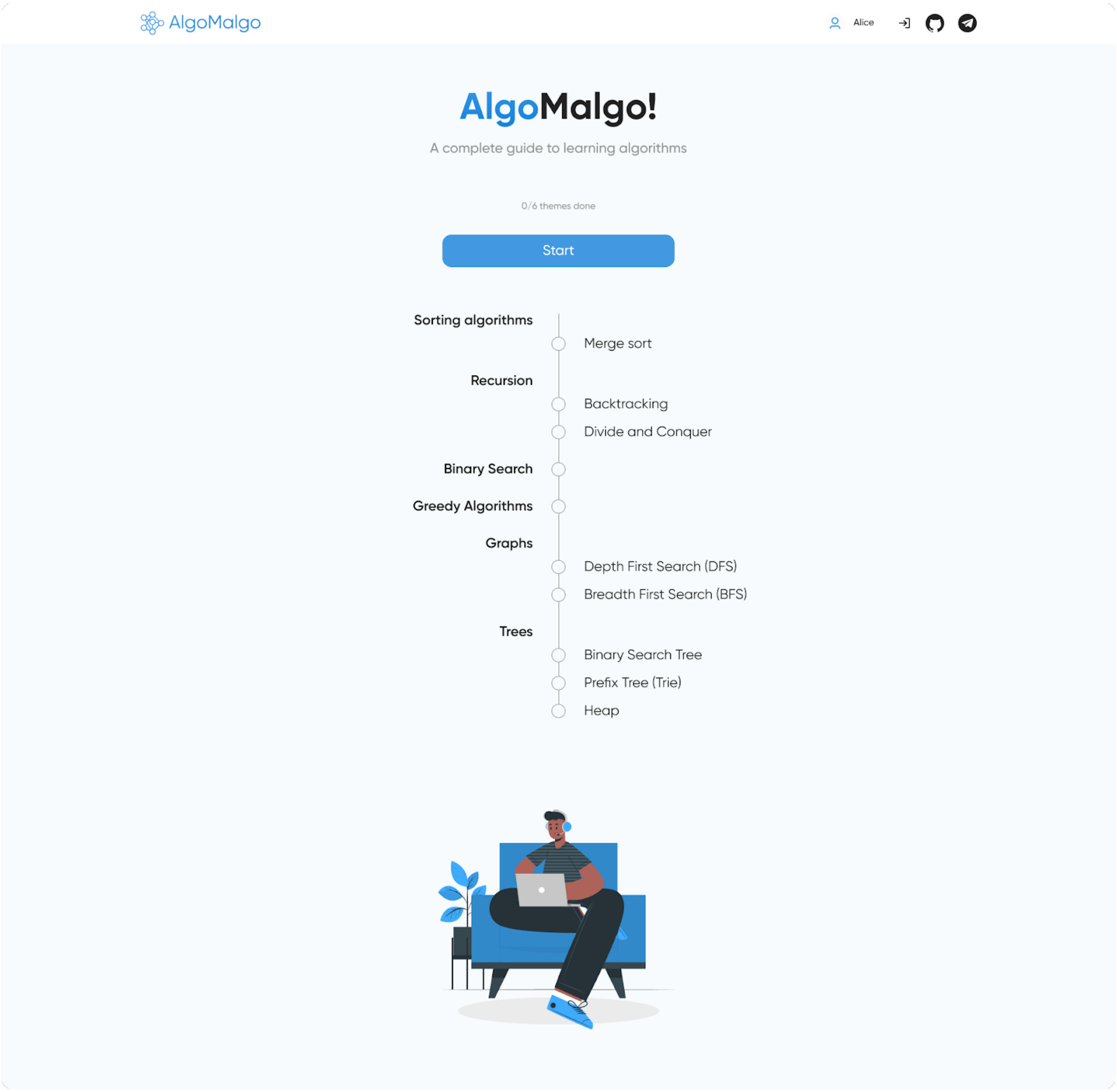
Password

••••••••

Repeat password

alexa123

Figure 5.3: Registration



AlgoMalgo!

A complete guide to learning algorithms

0/6 themes done

Sorting algorithms
- Merge sort

Recursion
- Backtracking
- Divide and Conquer

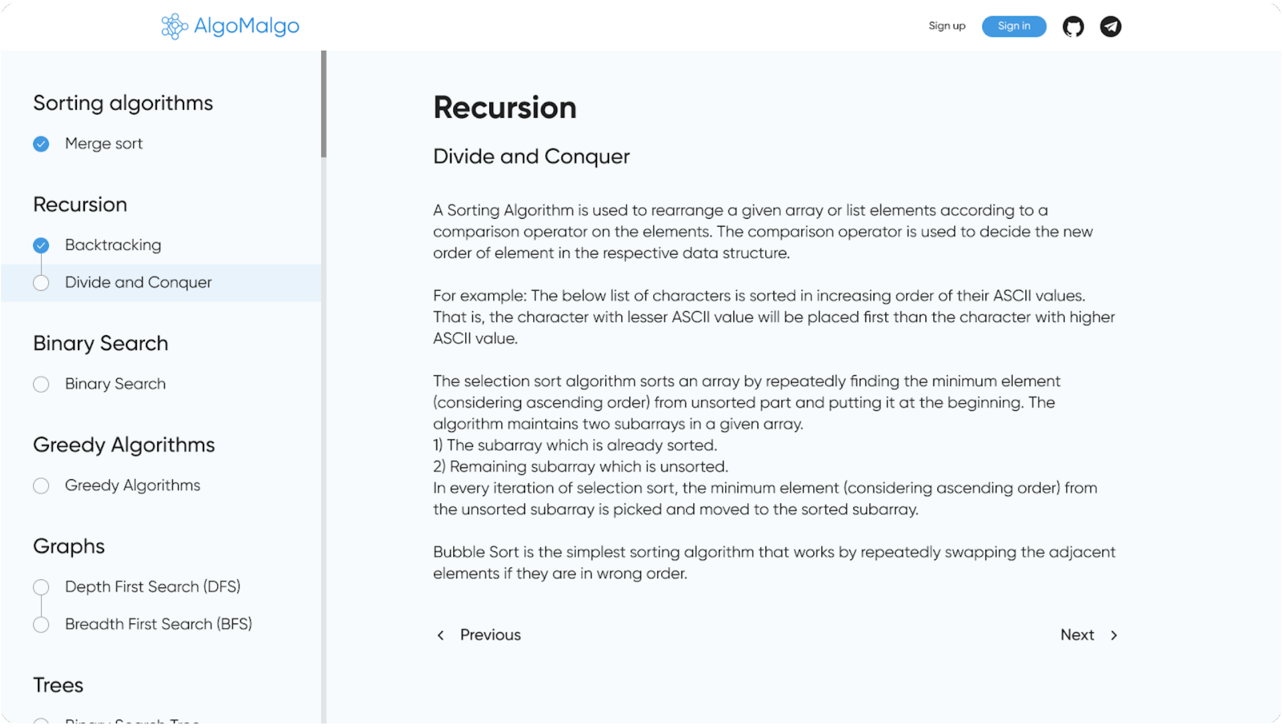
Binary Search

Greedy Algorithms

Graphs
- Depth First Search (DFS)
- Breadth First Search (BFS)

Trees
- Binary Search Tree
- Prefix Tree (Trie)
- Heap

Figure 5.4: Home Page



Recursion

Divide and Conquer

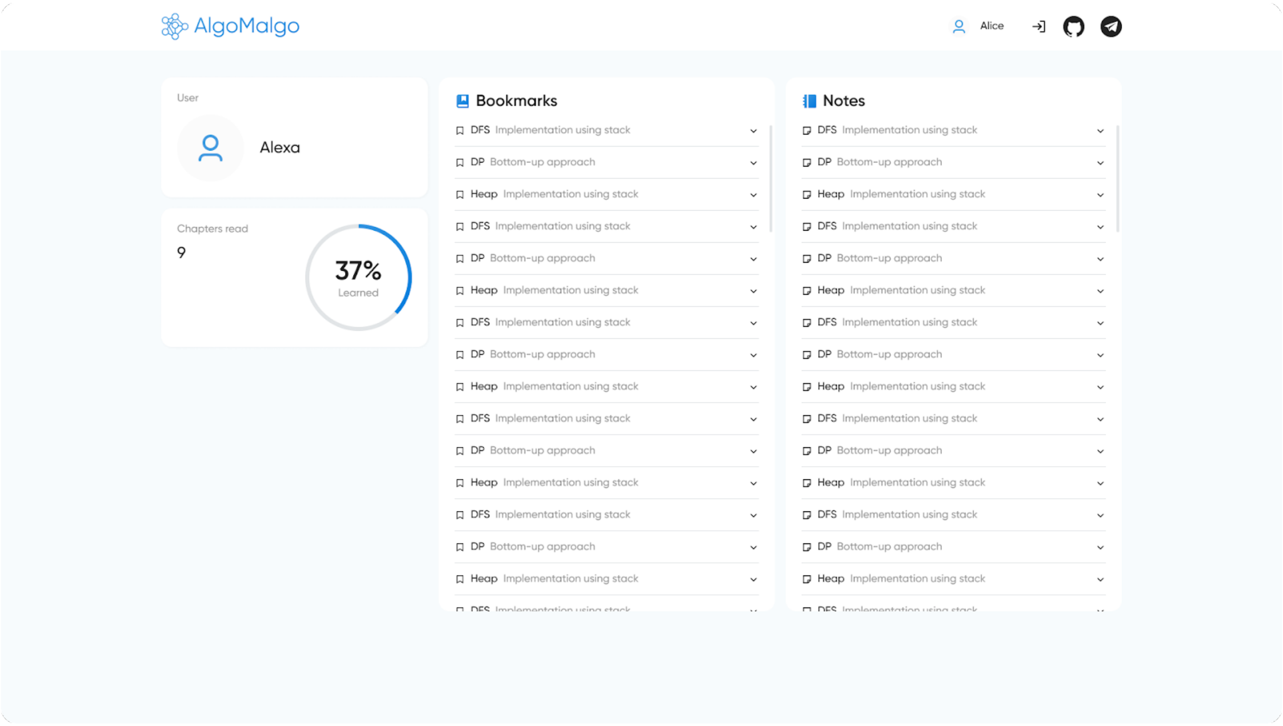
A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

For example: The below list of characters is sorted in increasing order of their ASCII values. That is, the character with lesser ASCII value will be placed first than the character with higher ASCII value.

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.
1) The subarray which is already sorted.
2) Remaining subarray which is unsorted.
In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Figure 5.5: Content



User
Alexa

Chapters read
9

37%
Learned

Bookmarks

- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack

Notes

- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack
- DP Bottom-up approach
- Heap Implementation using stack
- DFS Implementation using stack

Figure 5.6: Dashboard

5.0.3 System Architecture

System architecture of AlgoMalgo will be based on microservice, due its scalability and flexibility. Thus, with the increasing demand for our service, system design of website will change. As a main cloud computer provider AWS will be used, because of its cost-efficiency, flexibility and scalability.

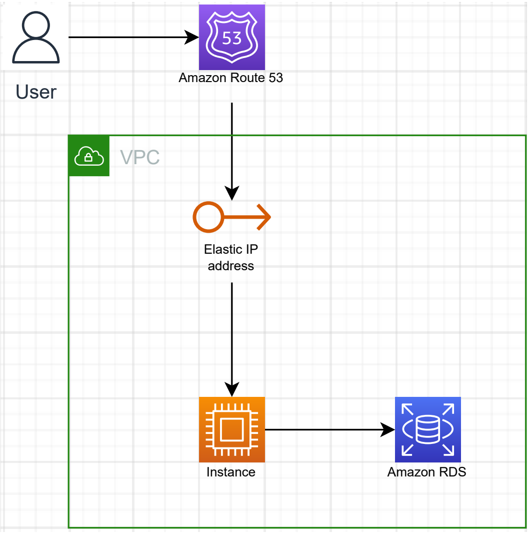


Figure 5.7: 100 users

In the above diagram, architecture is focused to embrace the number of users from 100 to 1000. As a main components Amazon Route 53, Elastic IP address, Instance and Amazon RDS will be used. All of them will be under virtual private cloud (VPC).

- Amazon Route 53 component will be an extremely reliable and cost effective way to route end users to Internet applications by translating domain names to the ip address of our instances.
- Elastic IP address is used to mask the failure of an instance or software by instantly remapping the address to another instance in the user's AWS account.
- Instance is a server where app will be hosted
- Amazon RDS is a highly efficient solution for our relational database needs

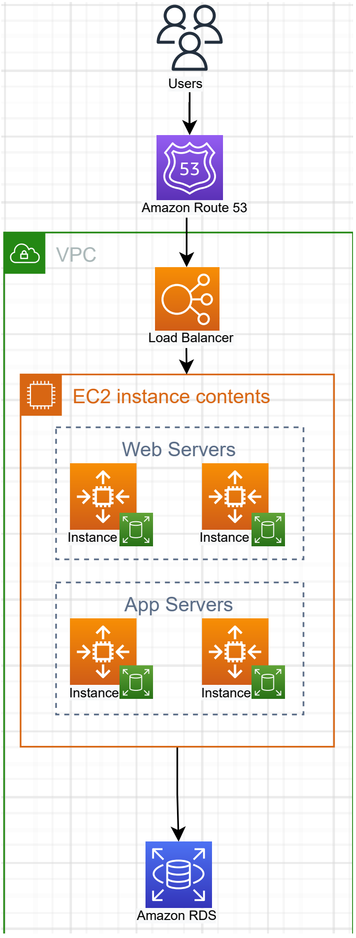


Figure 5.8: 1000 users

- EC2 instances will be used instead of simple instances, because of its autoscaling capabilities. This feature will be helpful, in the situations, when the number of users will exceed the normal expectations. Thus, in days of extreme load, our application will remain healthy

- Elastic Block Stores is a green component attached to each EC2 instance in the above diagram. It serves as a cache block to each EC2 instance, so that service will remain fast.

- Load Balancer is a key component in every system that has more than 10000 users. This entity used to evenly distribute the load across existing EC2 instances to remain stability of the existing system.

To maintain the number of users from 1000 to 50000, Load Balancer, EC2 instances will be used. Furthemore, to evenly distribute the load, application logic and web logic will be hosted on different servers.

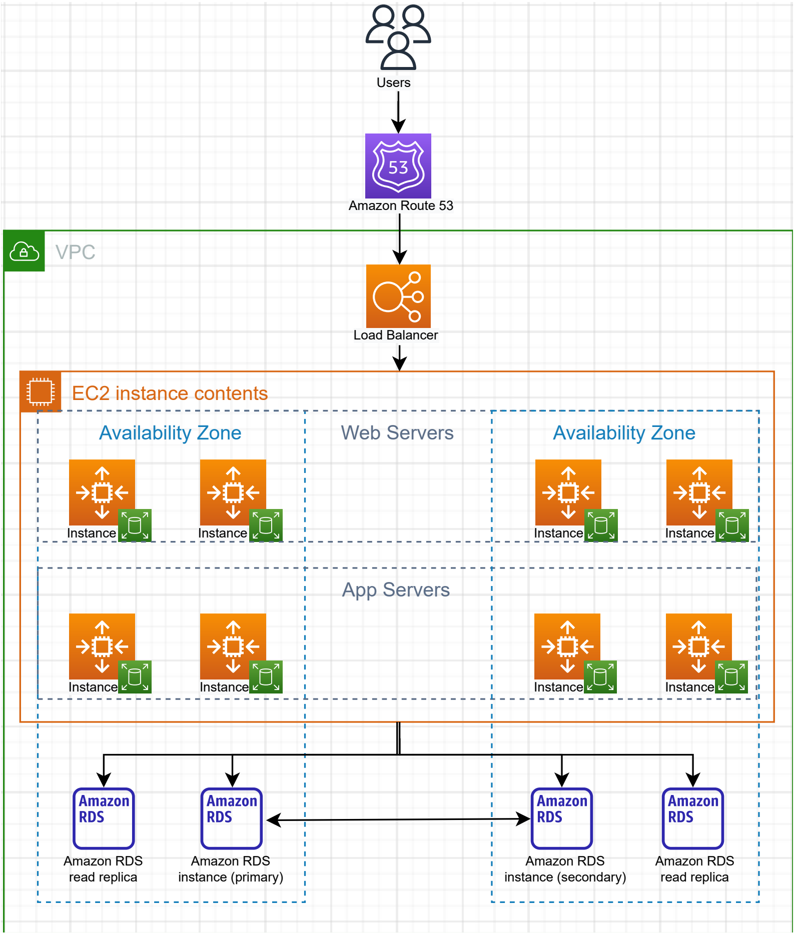


Figure 5.9: 1000 users

Diagram above, indicates a good system architecture that could maintain the number of users from 50000 to 200000. Key components of the new system are RDS read replicas, RDS primary and secondary instances, and different availability zones.

- RDS read replicas used to lower the load from RDS primary replicas, so that primary RDS replicas will handle only write requests. In the application it is supposed that the read operation requests will outperform by the number write requests. When the number of users will increase, the number of read replicas will increase proportionally.

- Availability Zone is a specific data point where EC2 instances and RDS components will be hosted. More Availability Zones indicate different data points, which will be located in different areas, so that users will not experience delays due their different locations across the country.

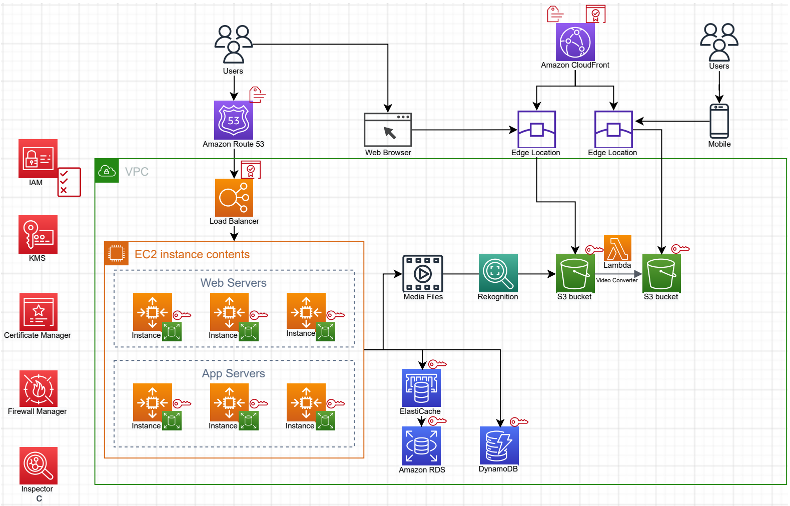


Figure 5.10: 10000 users

System architecture shown above designed to support the number of users from 200000 to 1 million. There are many new components such as: IAM, KMS, Certificate Manager, Firewall Manager, Inspector, Rekognition, S3 buckets, Lambda, ElastiCache, DynamoDB, EdgeLocation and Amazon CloudFront. Services as IAM, KMS, Certificate Manager, Firewall Manager and Inspector responsible for security measures. They provide great security out of box, which will be essential for a service having around 1 million users. Different availability zones, Database read replicas not shown in this diagram to keep it clean.

- Identity Access Manager is IAM in short. It is a primary service for managing all accesses of users in our AWS system. Accesses, authentication, authorization, restrictions managed by IAM. It will be used to give different permissions to a particular set of users, to manage an entire architecture.

- Key Management Service is KMS in short. It is a primary service responsible for all encryptions of data across the system. Services like DynamoDB, AmazonRDS, ElastiCache, S3 buckets and elastic block stores will be affected by KMS.

- Certificate Manager will be mainly responsible for providing SSL certificate to make secure HTTPS connections to services like Load balancer and Amazon CloudFront.

- Firewall Manager's key responsibility is preventing different web site attacks including cross-site scripting, SQL injections, DDoS attacks, etc. This application firewall will secure such components as Amazon Route 53 and Amazon CloudFront.

- Inspector scans existing machines for the purpose of finding any known vulnerabilities, so that developers will fix them.

- Rekognition serves as a content filter, which will find out inappropriate objects in an image and will prevent this image being uploaded to S3 bucket.

- S3 bucket is an unlimited external storage, content on that storage could be accessible from any endpoint on the internet. In our system S3 bucket is mainly used for storing user media files, like images and videos.

- Lambda is a serverless service of Amazon, which will be used to convert one media format to another. It is expected that having 1 million users, that some of them will access AlgoMalgo through mobile phones. To maintain good user experience, it is essential to convert media files specifically for their devices.

- ElastiCache is a cache service for RDS databases. It comes with Redis and Memcached engines in it. Using ElastiCache we will reduce database calls, so that data that is accessed often will not heavily load RDS.

- DynamoDB is a NoSQL database solution. It will be essential for storing BookMarks and Notes logic. Currently, BookMarks and Notes stored in RDS, however having 1 million users it will be a wise decision to store these data on NoSQL database.

- Edge Location is a nearest location, where static content will be cached.

- CloudFront is a content delivery network (Cache), that is used for caching static content through accessing the nearest edge location. It is a good solution for caching any media content like images and videos. Because our system consists of detailed video explanations of algorithms and has various images explaining step by step solutions, it would be a great idea to have such service.

Chapter 6

Conclusion

The purpose of this thesis was to understand how promising and effective gamification in education is. Based on the amount of interest in this topic and successful applications in various fields, it can be assumed that gamification has good potential in the future. We examined the key elements of game elements that are successfully used in various fields of education. We analyzed the gaming industry and tried to find out the reason why so many people play different types of games.