

**MotoProg**

**Project Number:**

**22-2-D-28**



**Advisor:**

**Naomi Unkelos-Shpigel**

**Liran Haim 308292382** [**Liranhaim77@gmail.com**](mailto:Liranhaim77@gmail.com)

**Sapir Dayan 313605776** [**Sapirdayan@gmail.com**](mailto:Sapirdayan@gmail.com)

**Content**

0. ***Abstract***……………………….………………………….…..………………………..3

1. **Introduction**………………………….…………………………..……………………3

1.1 Defining the problem………………………….………………………………3

1.2 Scope of the project………………………….……………………………….3

1.3 Document Review………………………….…………………………………3

2. **Background and Related Work**………………………….………….…………….…4

2.1 Base Definitions………………………….……………………………………….....…4

2.2 Background………………………….……………………………………..…...4

2.3 Related Works………………………….……………………………………….5

2.3.1 Code Combat………………………….………………………………..5

2.3.2 CodeEasy………………………….……………………………………5

2.3.3 CoderByte………………………….……………………………………5

3. **Expected Achievements**………………………...………………………………….…6

3.1 Project Goals………………………….………………………………………...6

4. **Engineering Process**…………………….….………………………………………....6

4.1 Process………………………….………………………….…………………...6

4.2 System………………………….………………………….……………….…...7

4.3 choosing questions….………………………….…………………...…………7

4.4 checking motivation….………………………….…………………...…………8

4.5 The Architecture….………………………….…………………...…………….8

5. **Product**. ………………………….…………………………...…………………………9

5.1 **Activity Diagram** ………………………….…………………………………...12

5.1.1 Login……………………………………………….………………...12

5.1.2 Register……………………………………………….……………..13

5.1.3 playing……………………………………………….……………..13

5.2 **Package Diagram** ……………………….……………………………………...14

5.3 **Screens** ……………………….………………………………………….……..15

6. **Evaluation/Verification Plan**………………………….……………………………...25

7. **User Guide**.……………………….…………………………...……………………..…27

8. **Programmer Guide**.…….…………………………...………………………….…..…27

9**. Results and conclusions**.…….…………………………...……………………..…30

10. **References**………………………….…..…………………….……………………….31

**Abstract**

The use of Gamification which is adding game mechanics into nongame environments has

greatly increased during the last years. At the same time, technology advances have opened

new spaces and possibilities for the field of computer-based edutainment-education in the

form of entertainment - where learners can achieve their learning goals while having fun.

In our project we develop web interface that contains a learning game on a programming language, principles, and in-depth theoretical ideas in the world of programming. The system explores the motivation of the user while he is using the system.

This will be done through a combination of gamify elements.

During development we use advanced technologies that allow us to establish a user-friendly and up-to-date interface.

1. **Introduction**
   1. **Defining the problem:**

When students do not find interest in the study material, they may be in a state of significant lack of knowledge [2]. These days we are all distracted, smartphones and the internet are catching our attention.

Although, there are many platforms on the Internet that the user can learn through, but during the learning process the user can lose interest and motivation. This will mean that the learning process may not be effective, the learner will not feel obligated to the learning process.

* 1. **Scope of the project:**

'Gamification’ has become a prominent method for encouraging habits and enhancing motivation and engagement. Game approaches lead to a higher level of commitment and motivation of users to activities and processes in which they are involved.

“MotoProg” is a web application that was designed to solve this problem and allow the user to gain knowledge in the software world while checking the user's motivation during his learning process.

This system is the ultimate tool to teach software principles to users and improve

their capabilities while the system checks their motivation through the process.

* 1. **Document Review:**

At the beginning of the project book, we will present the background for choosing the project theme, the software, and services currently available in the market that provide a platform for learning programming language, principles, and in-depth theoretical ideas in the software world.

In addition, we will see the technologies we use to implement the solution we offer. We will then present the achievements we expect to achieve. The main chapter of the book details the product development process that shows what we did along the way to understand how we are improving the options available today and what actions we have taken during the work. And we will see the initial characterization of the product which includes diagrams describing the structure of the system and the interfaces and logic between them.

Finally, we will detail the system requirements and present a detailed testing plan.

1. **Background and Related work**
   1. **Base Definitions:**

Gamification

Gamification is defined by Werbach

and Hunter as “The use of game elements and game-design techniques in non-game

contexts'', it means that the term "Gamification" is the use of game elements and design in non-game contexts [3].

For example, in-game principals and themes such as acquiring virtual ‘points' or other

currency and completing a series of tasks or activities to advance to the next level, may be used in contexts other than gaming to provide fun and stimulation for the learner.

Gamification can also be defined as a set of activities and processes to solve problems by

using the characteristics of game elements [2].

WOLF questionnaire

The WOLF questionnaire is an instrument for the measurement of flow experiences at work. It is an initial set of 16 elements that were used to operationalize

flow. This collection was trimmed to 13 questions based on exploratory

component analysis and reliability analyses. Absorption (4 things), job enjoyment (4 items), and intrinsic work motivation (5 items) are the three flow dimensions that are measured. On a seven-point scale, respondents stated how often they encountered elements of flow (1= never, 7 = constantly). The check of the user's motivation will be through this questionnaire [1].

* 1. **Background:**

Today, Due to the rapid growth of information and communication technologies, the education environment has been enriched with technologies and become more diverse [3].

In Software engineering there are topics that are less interesting or alternatively more

difficult to understand, this can negatively affect the motivation of learners [4].

In our days there are many systems that can teach the user on any subject and particularly in programming and in software engineering and involved gamified tools

but most of these applications/webs don’t check the user's motivation through the learning process [4].

* 1. **Related works:**

We analyzed the current market and found a list of websites that allows learning

programming in a gamified way.

* + 1. **Code Combat -**

"Code Combat" is a learning a programming language web, that involves video game. To complete a level, you need to type pieces of code provided within the game. Before the game starts, you choose the language you wish to learn

There is a large use of Gamification in this website. The user gains points through the activities. There is a Progress screen with a link to experience Ranking leaderboard. Large number of levels-400 levels, offer a good opportunity to develop advanced skills and drill developed skills.

The platform doesn't check the motivation of the user using the website [5].

**CodeEasy:**

“CodeEasy” is a web that provides adventurous learning path. The narrative is to save the world from a machine invasion. Each chapter in the story will teach you fundamental or fundamentals related to C#. The user can also compare his answers with the other coders’ answers.

The user can use their Slack group to communicate with fellow coders. There is a lot of practice and engagement on this platform. There is an adventure story in Codeasy, making it addictive and engaging. They equip every lesson with several practical assignments.

The platform doesn't check the motivation of the user using the website [6].

**CoderByte:**

“CODERBYTE” is an online learning website that includes beginner-friendly tutorials and courses about programming languages, It also has a bunch of practical coding challenges you can complete. The user can execute code directly on their site. There are tasks that users perform and progress towards defined objectives. There is an option to discuss with other users about solutions [7].

The platform doesn't check the motivation of the user using the website.

All the websites we have presented here involved gamification tools, but do not check the user's motivation through the learning process.

We offer a web that includes either a learning process that involves gamification tools and checks the user's motivation.

1. **Expected Achievements**

In our project we expect to develop a user-friendly website, which will be adapted and up to date for several platforms (such as computer, cell phone, tablet). Users will be able to register to the system and to choose a category which they want to learn. Through the learning process the system will check their motivation.

**3.1. Project Goals:**

1. Our main goal is to make the software learning process intuitive and more inviting.
2. By involving WOLF questions in the quizzes, we check the user's motivation through the learning process.
3. In order to enhance user engagement, we've incorporated gamification elements like points, hints, and a 50:50 option into our system. Section 4.2 provides a more detailed explanation.
4. Develop a responsive website that will be adapted for any platform such as cell phone, computer, tablet, etc. so that the user can learn from any device in any time.
5. **Engineering Process**

**4.1. Process:**

First, we analyzed the problem we chose to solve, conducted a survey with stakeholders who are developers from the industry, and understood from them what are the important emphases we want to focus on and what are the needs of our system. We then sat down and defined in depth what the requirements of the system were.

We have prepared an orderly work plan that includes all the steps, from the beginning that contains the learning process and later on our development process until we finally reach a final result that meets the desired goals that we have set for the system. After research, we realized that the most effective and correct approach for our project to develop the system is in the Agile approach, an iterative method that emphasizes the need for work in stages with continuous planning and continuous learning.

As a first step, we had to understand the platforms and exchanges that exist today in the market, understand their advantages and disadvantages and how we can preserve the advantages and in addition to improve and reduce their disadvantages to provide unique ideas to our system.

**4.2. The process of the system:**

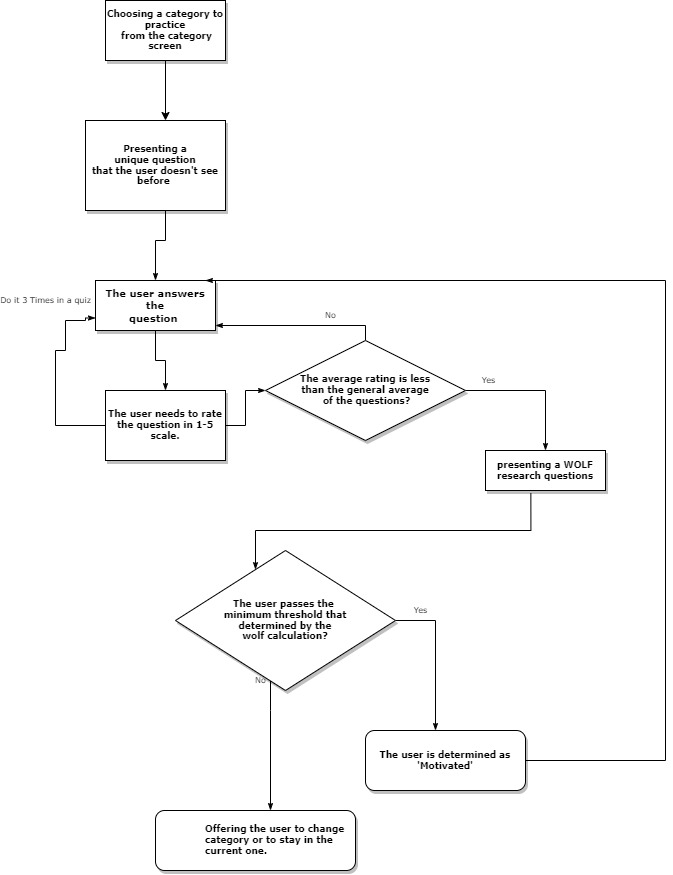
* Our system is divided into categories, each category presents questions in the specific topic. Each question had a unique ID.
* Our platform checks the motivation of the user during his learning progress. This check done by presenting the user 3 questions from the Wolf questionnaire.
* To increase the user's motivation through the process we use "Gamification Mechanisms". In our system it is done by:
  1. **Points-** are accumulated because of answering correct on the questions. The system presents the user his score during all the process, the user can see it during his learning process.
  2. **Ranking** -the questions in each category
  3. **Hint –** when the user tries to answer a question, he can use the icon light that appears on the question screen, clicking on the icon will remove an option from the answers and advance the user to the solution.
  4. **50:50** - In every question screen there is an option to 50:50. The 50:50 is presented by the icon in the right side of the screen, clicking on the icon will remove half of the options from the answers.
  5. **users** are all participants – employees or clients (for companies), students (for

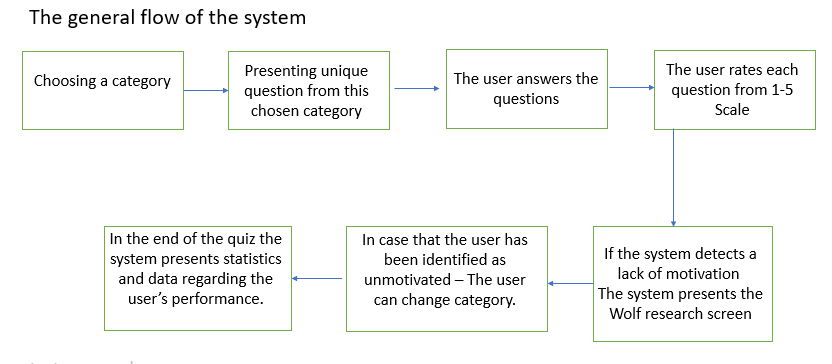
educational institutions).

* Our system offers the user the opportunity to give feedback on each question in

the chosen subject.

* The system uses the WOLF questions in the quiz to evaluate user motivation based on their answers. More information on this process can be found in section 4.4, which details the motivation calculation algorithm.

**System Flow Diagram:**



**Fig 1:** Flow Diagram

**4.3. The algorithm for choosing questions:**

1.First the user chooses a category to practice from the category screen

2.Presenting question to the user that he doesn't see before, the question is from

the specific category in a random way, the question is chosen from the questions

that received the highest rating [the rating for each question determined by all the

users that rate this specific question].

3.The user answers the question, after the user answers the question, he needs to

rate this specific question.

4.After the user answers about 3 questions in the specific quiz from the chosen

category – the algorithm checks if those 3 questions have an average rating that is less

than the general average of the questions – if not -the algorithm returns to step 2, if yes –

the algorithm moves to step 5.

5. The research question that is presented to the user is chosen randomly from each

field: absorption, work enjoyment, intrinsic motivation.

For every research question there is a range between 1-7, if the user passes the

minimum threshold that determined by the wolf calculation, we can determine that

the user is motivated and returns to step 2

If the user doesn’t pass the min threshold – the algorithm offers him to change a

category

6. If the user does not rate the question below the overall rating assigned to this

question – returns to step 2.

**4.4. The algorithm for checking the motivation of the user:**

We use the tool called "WOLF" in our system.

The check of the user's motivation will be through this questionnaire.

We changed the wording of the questions to match them to our system.

* If the system detects that the rating of the questions that the user answers in sequence is lower than the general rating, then the system presents a research question that includes three statements from the three different fields.
* The user needs to rate the statements.
* If the rating of the statements is lower than the threshold rating that was set in the algorithm research, then the system suggests the user to move to a different category.
* If the rating is equal to or higher than the threshold rating, the system offers the user to continue with the same quiz.
* The user will need to rate the statements in a specific subject, the way he will do it

will be the same as they did in the research – on a 5 points scale. Through the user's feedback the system analyzes and get a perspective about his motivation in the specific quiz / pool of questions.

**4.5. Software Architecture:**

The architecture in our system divided to three parts:

**Client Side:**

The First part was the Client-Side, it was built using React based on node js.

React is a JavaScript library for building user interfaces. It is widely used for creating web and mobile applications.

React allows us to build reusable UI components, making it easier to create our system as complex and large-scale.

React provide us to follow the unidirectional data flow, which makes it easier to track and manage our system state.

Cross-platform: React Native allows us to use the same codebase to make our system cross-platform. Choosing React in the client side of our system makes it fast, scalable, and easy to maintain [8].

Node.js is a JavaScript runtime that allows developers to run JavaScript on the server side and the client side, we use it for the client side.

Node.js is built on the V8 JavaScript engine, which is the same engine used in Google Chrome. This provides Node.js with a high level of performance and allows us to handle a large number of concurrent connections.

Node.js is designed to be asynchronous and event-driven, which makes it well-suited for building high-performance, real-time applications, such as chat and gaming applications which is found suitable for the essence of our system [9].

**Server Side and Database:**

The second part is our Server-Side and Database, we use 'Serverless'

Serverless refers to a cloud computing execution model in which the cloud provider is responsible for executing a piece of code by dynamically allocating the resources.

With serverless, we only needed to provide the code and the cloud provider took care of executing it, scaling it, and providing an infrastructure to run it. we only charged for the exact number of resources and time that the code is using.

Serverless automatically scaled our application to handle changes in traffic, so we didn’t need to manage additional servers. Serverless allowed us to build and deploy the system quickly and easily, without managing underlying infrastructure. Serverless provided security features, such as automatic patching and updates, encryption, and more, which helped us keep our application secure. Serverless architectures are event-driven, which means that our code only runs when a specific event occurs, such as a user rating a question or answering a research question. This leads to a more efficient use of resources.

We use Firebase, it is a platform that provides a variety of services for building and managing web and mobile applications, one of them is the Firebase Realtime Database, which is a cloud-hosted NoSQL database that allows developers to store and sync data across multiple clients in real-time. We used this database in a serverless architecture, which means that the cloud provider is responsible for executing the code and allocating the resources needed to run it, while we only needed to provide the code and the data [10].

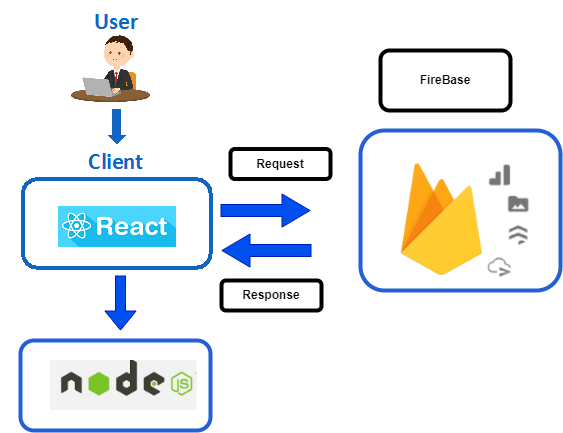
**Cloud:**

The third part is using a cloud that contains our app. We use 'Netlify' as a cloud for our app. Netlify is a popular cloud-based hosting and deployment platform for web applications, including React projects. It provides a fast, secure, and scalable infrastructure for deploying web applications with ease.

With Netlify, we deploy our app by connecting it to our Git repository. Netlify automatically builds and deploys our project every time we push a new commit, ensuring that our app is always up-to-date and running smoothly.

One of the key benefits of using Netlify is its continuous deployment feature, which eliminates the need for manual deployment and allows us to focus on developing our app.

Overall, Netlify is a powerful and flexible hosting and deployment platform that helps us streamline our app development and deployment workflow.

**Architecture Diagram **

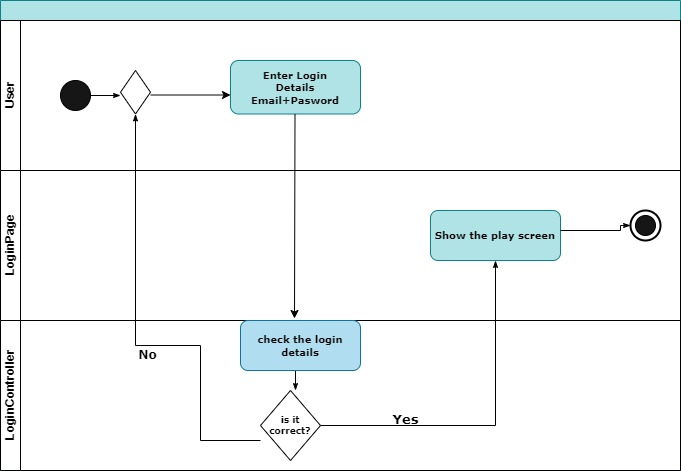
**Fig 2:** Architecture Diagram

1. **Product**

**5.1 Activity Diagram**

**5.1.1 Login**

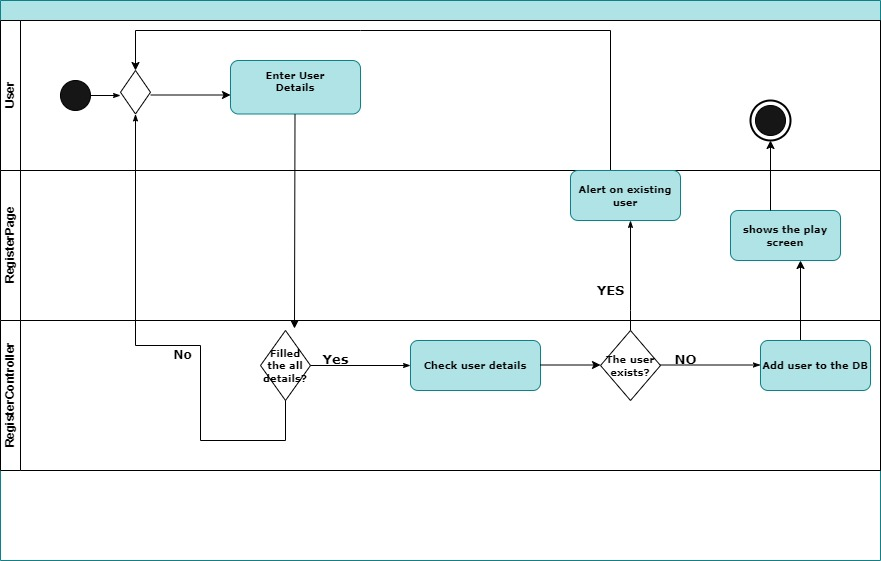
1. The user inserts the login details.
2. The system checks if the details are correct.
3. If the details are incorrect moves to step 1.
4. If the details are correct show the play screen.



**Fig 3:** Login Diagram

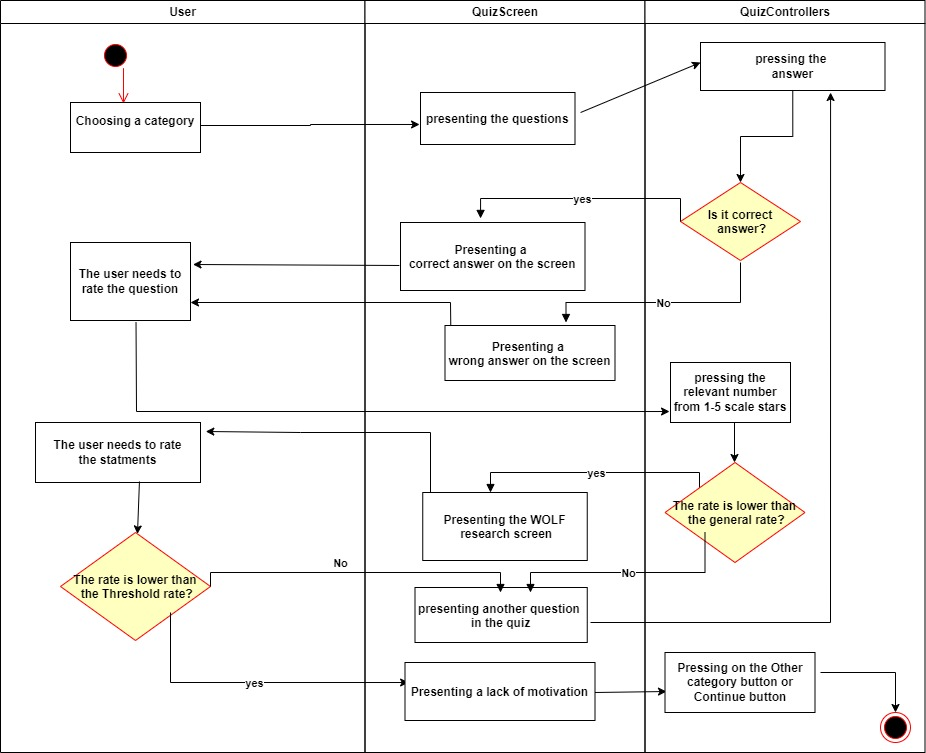
**5.1.2 Add new User - Registration**

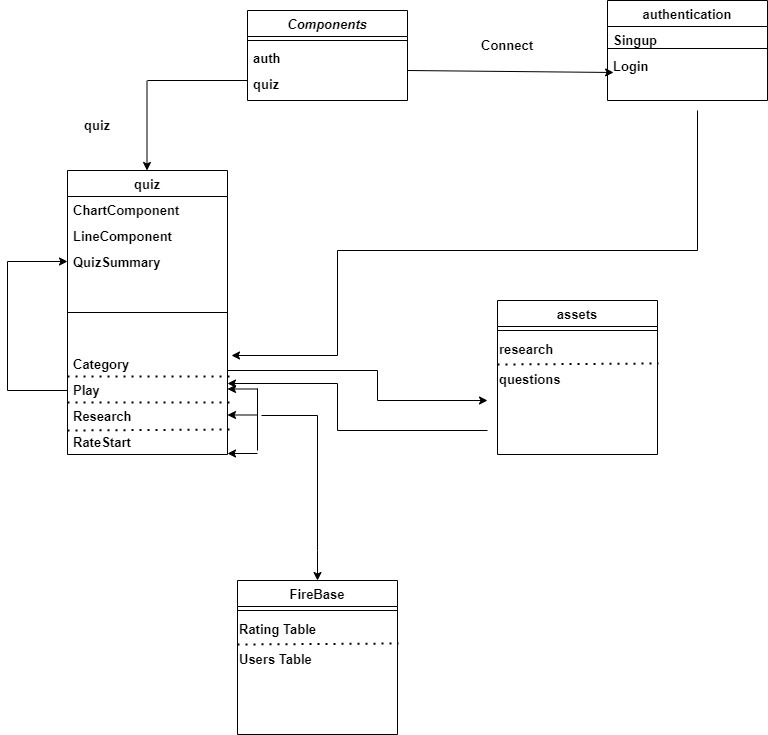
1. The user inserts his details.
2. The system checks If all fields are filled.
3. If details are missing pass to the step 1.
4. The System check if this user already exists in the Database.
5. If the user doesn’t exist pass to step 7.
6. If the user exists, the system alerts to user about existing user.
7. The user is added to the user's table in the DB.



**Fig 4:** Register Diagram

**5.1.3 Playing the game**

****

**5.2 Package Diagram**

**Fig 6:** Package Diagram

**5.3 Screens -**

Graphical user interface, application

Description automatically generated**The Home Page:**

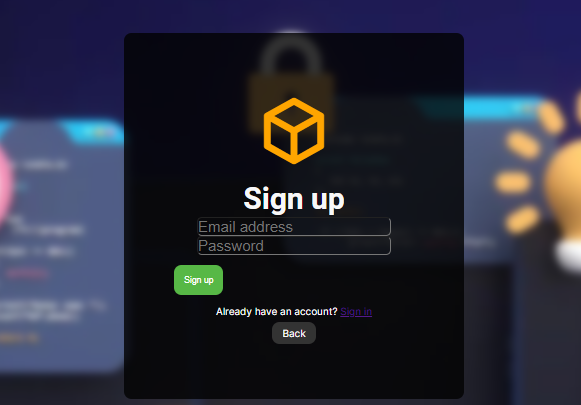
**Fig 7:** The Home page

Graphical user interface, application

Description automatically generated**The Login page:**

**Fig 8:** Login Page

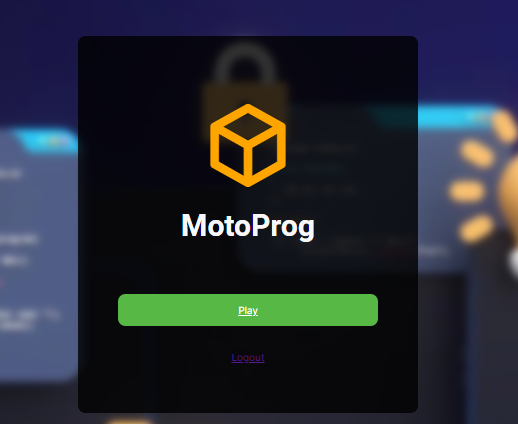
**The Register Screen:**



**Fig 9:** Register Page

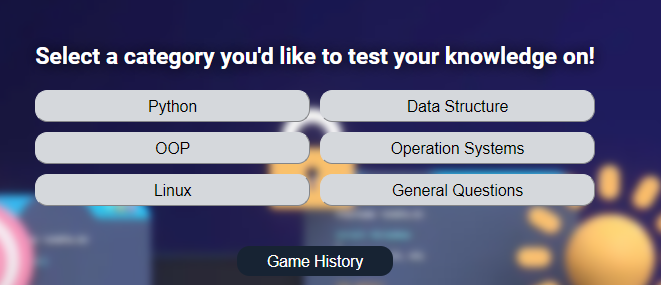
**The Play Screen:**

After filling the details, the system will provide the user to move to the 'play' screen.

**The Categories Screen:**

**Fig 10:** Play Page

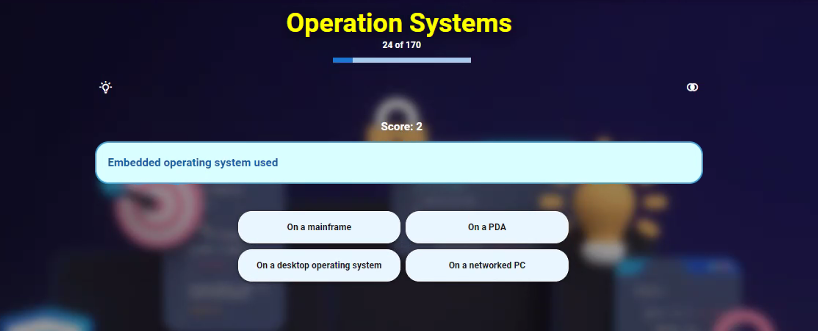
After the user moves to the 'play' screen, the system will show him the 'categories' screen



**Fig 11:** Categories screen. 

**The Quiz Screen:**

After the user chooses a category, the system enters him to the quiz screen



**Fig 12:** Quiz screen

**Question review:**

After each question, the user needs to rate the question from 1 – 5.

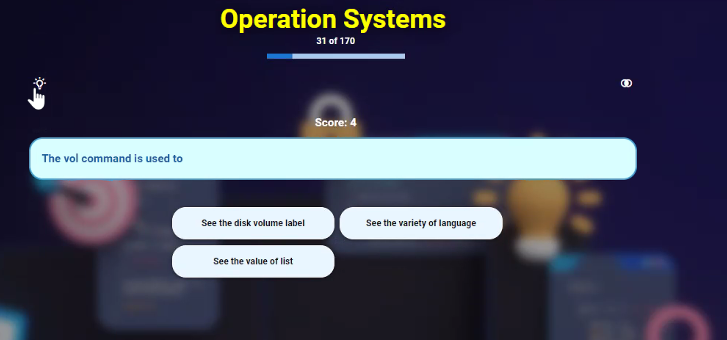
A picture containing text, screen

Description automatically generated

**Fig 13:** Question Review

**Hint Option:**

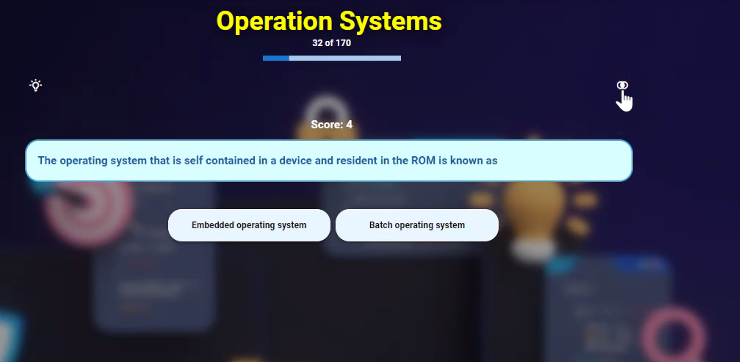
In every question screen there is an option to get a hint. The hint is presented by the light bulb icon in the left side of the screen. After the user get a hint, the number of the answers options is reduced. The number of times the user can use the hint option is not limited in the questionnaire but is limited to one time for the same question**.**



**Fig 14:** Hint option

**50:50 Option:**

In every question screen there is an option to 50:50. The 50:50 is presented by the icon in the right side of the screen. After the user choose to use this option, the number of the answers options is reduced to two options. The number of times the user can use the hint option is not limited in the questionnaire but is limited to one time for the same question



**Fig 15:** 50:50 option

**The Research questions screen:**

The system shows the user research questions. Those questions based on the 'WOLF' research. The user needs to rate the statement, and the system identifies if the user is motivated or not.

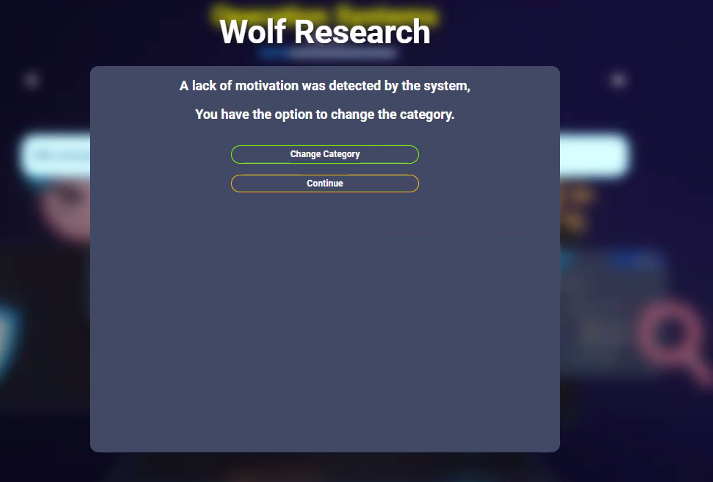
Graphical user interface

Description automatically generated

**Fig 16:** Research questions Page

**The WOLF Research screen:**

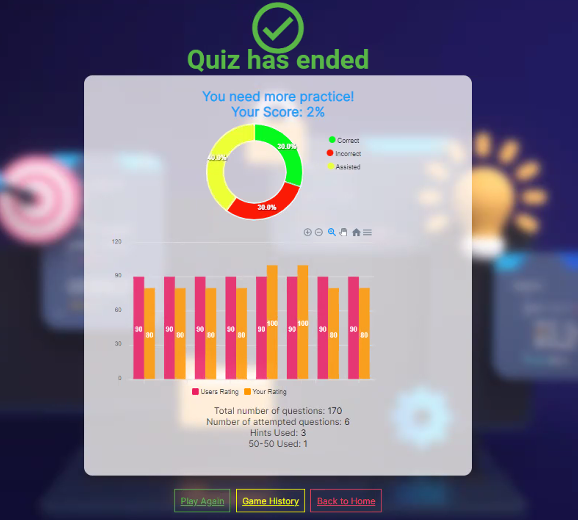
After the user answers on the three 'WOLF' research questions In case that the system identifies that the user is not motivated – it offers him to change category or to continue with the current one.



**Fig 17:** WOLF Research Page

**The Summery Screen:**

After the ends the quiz or press 'quit' during a quiz, the summery screen appears. The user can see there graphical and textual display of the ratio of the questions that he answered correctly or incorrectly during the session. In addition, a graph that shows his ratings in relation to other users who answered the same questions.

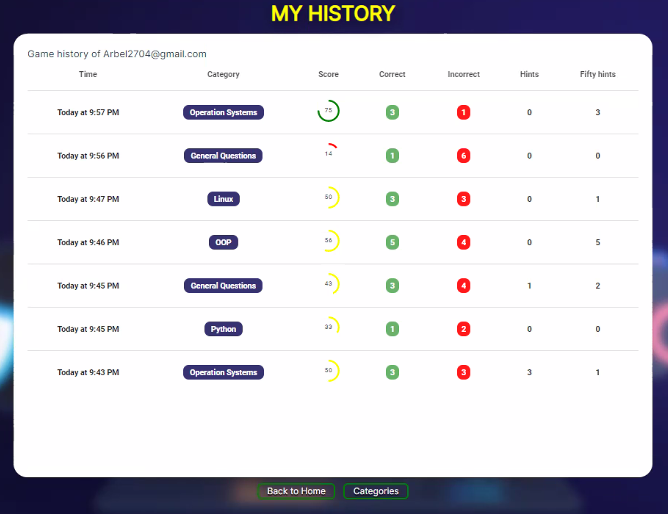


**Fig 18:** Summary Page

**Game History screen**

The user can see his games history in the screen.

The screen represents the user's score in the different categories.



**Fig 19:** Game History screen

1. **Evaluation/Verification Plan**

To evaluate the system, we will perform evaluation and verification tests.

**Testing plan**:

|  |  |  |
| --- | --- | --- |
| **UserName** | **password** | **Exists in Database** |
| Liran77@gmail.com | Liran77 | Yes |
| Sapirdayan@gmail.com | Sapirdayan31 | No |
| Yuvaldabush78@gmail.com | Yuvi888 | No |

**Register:**

|  |  |  |
| --- | --- | --- |
| **Test Name** | **Description** | **Expected result** |
| Register success | UserName: Sapirdayan@gmail.com password: Sapirdayan31 click: Sign up | user is added to the Firebase and the user moves to the Play screen |
| Register short password | UserName: Sapirdayan@gmail.com password: 11  click: Sign up | **Our app displays the error:**  Password should be at least 6 characters |
| Register username exists | UserName: Liran77@gmail.com password: Liran778  click: Sign up | **Our app displays the error:**  This email is already existing |
| Register empty fields | UserName: Sapirdayan@gmail.com password:  click: Sign up | **Our app displays the error:**  This is a mandatory field |
| Register corrupted email | UserName: @gmail.com password: sap1115  click: Sign up | **Our app displays the error:**  The email is incorrect |

**Login:**

|  |  |  |
| --- | --- | --- |
| **Test Name** | **Description** | **Expected result** |
| Login success | UserName: Liran77@gmail.com password: Liran77  click: Login | The app moves to the play screen**.** |
| Login user doesn’t exist | UserName: Yuvaldabush78@gmail.com password: Yuvi888  click: Login | **Our app displays the error:**  User not found. |
| Login wrong password | UserName: Liran77@gmail.com password: Liran779  click: Login | **Our app displays the error:**  Wrong Password**.** |
| Register empty fields | UserName: Liran77@gmail.com password:  click: Login | **Our app displays the error:**  This is a mandatory field |
| login\_register | Click: Not Account yet? sign in | **Our app switches to the Sign in screen.** |

1. **User Guide**

Software Environment:

* Windows Software
* React = 18.2.0 version
* Firebase = 9.15.0 version
* Node.js = 16.19.0 version
* IDE = VSCode

**User Instructions:**

Press the link in order to get into the system:

<https://stalwart-rolypoly-fb9fd0.netlify.app/>

The system works as a cross-platform. The user can get in from any device.

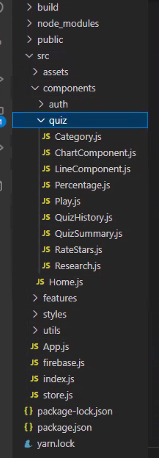
1. **Programmer Guide for MotoProg App**

**8.1 Purpose and Scope**

This guide is intended for developers who will be working on this app. It provides an overview of the app's purpose and functionality, explains the folder and file structure, and documents the coding conventions and standards used in the project.

**8.2 Folder and File Structure**

The following is an overview of the folder and file structure used in our app:



**8.3 Running the App**

To run the MotoProg App on your local machine, follow these steps:

1. Clone the repository:

git clone https://github.com/liranhaim1/MotoProg.git

1. Install dependencies:
2. npm install

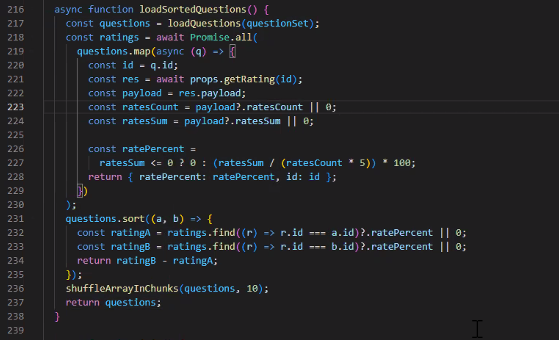
3- Start the development server: npm start

1. The app will be available at <http://localhost:3000>.

**Main Functions:**

**LoadSortedQuestions Function:**

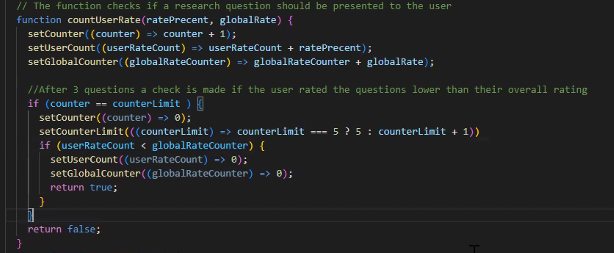
We load the questions from the set of questions and sort them by their rating.



**CountUserRate Function:**

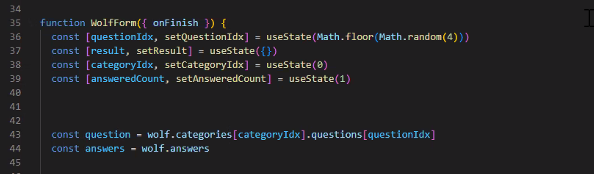
In this function we check if the rate of the questions of the user is less then the general rate of the other users in the system that answered on those questions.

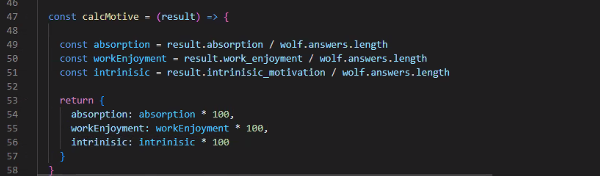
If the user rate is less than the general rate we present him the WOLF Research screen.



**WOLF Form Function:**

We choose the questions research in a random way.



We calculate the motivation, the calculation refers to the three parts in the WOLF Research.

**9. Results and conclusions**

In Our project we proposed a new platform to gain knowledge in the software world in a way that checks the user's motivation. The work was hard and intensive but very teaches and worthwhile.

In the phase A of the project, we dealt with the definition of the problem and characterize our system as a solution to this problem. As part of the process, we used the knowledge that we gain through the degree and use it for the different fields that existing in our system.

In the phase B, after we have finished the characterization part of the system, we started the process of developing the system.

**10. References**

[1] Bakker, A. B. (2008). The work-related flow inventory: Construction and initial validation of the WOLF. Journal of vocational behavior, 72(3), 400-414.

[2] D. J. Dubois and G. Tamburrelli, “Understanding gamification mechanisms for software development,” in Proceedings of ESEC/FSE 2013 (9th Joint Meeting on Foundations of Software Engineering), ser. ESEC/FSE 2013. ACM, 2013, pp. 659–662.

[3] Majuri, J., Koivisto, J., & Hamari, J. (2018). Gamification of education and learning: A review of empirical literature. Proceedings of the 2nd international GamiFIN conference, GamiFIN 2018. CEUR-WS

[4] Dal Sasso, T., Mocci, A., Lanza, M., & Mastrodicasa, E. (2017, February). How to gamify software engineering. In 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 261-271). IEEE

[5] Code Combat- <https://codecombat.com/>

[6] CodeEasy - <https://codeeasy.io/>

[7] CoderByte - <https://coderbyte.com/>

[8] REACT Guide

<https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>

[9] NodeJS Guide

<https://nodejs.org/en/docs/guides/>

[10] Firebase Guide

<https://firebase.google.com/docs/guides>

[11] GIT To Our Project

<https://github.com/liranhaim1/MotoProg>

[12] Our Project in the server

<https://motoprog.netlify.app/>