كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Information Systems Department

Your School

Prepared by:

[Engy Ashraf Azoz (20170118)]

[Rana Mohamed Atiea (20170225)]

[Rana Hisham Ragab (20170227)]

[Rahma Usama Hussein (20170218)]

[Donia Essam Mohamed (20170208)]

[Adly Mohamed Ahmed (20160239)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Information Systems Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

[Dr. Ahmed Yakoup]

July 2021


Your
School

Faculty of Computer and Artificial Intelligence
Helwan University.
Information System Department.

Supervised By:

DR. Ahmad El-Sayed Yakoup

Team Members:

Rahma Usama

Engy Ashraf

Donia Essam

Rana Hisham

Rana Mohamed

Adly Mohamed

INTRODUCTION:

First, our project is a School System. It is a mobile application which makes it easy for both children and their parents to find a suitable school. There are many people who are looking for a school every day, the parents have some specifications about the school they want so how can they find it? We also knew that some of the students miss them a year without finding any suitable school.

Second, there are many schools that want to increase the number of students they have but what can they do instead of marketing to their own organization?

Our application is considered an answer to these questions. It helps the children find a school and helps the school find its students. They can communicate with each other easily, the parents can find the nearest and the most suitable school for their children, they can see the whole instructions of the schools and what they provide, finally they can save their money and time.

The parents can create an account to make it easy for them to take look at the schools' profiles to take a decision.
The schools have the same property, they can create an account to receive any participants (Parents/Children).

DEDICATION:

This project is wholeheartedly dedicated to our beloved parents, who have been our source of inspiration and gave us strength when we thought of giving up, who continually provide their moral, spiritual, emotional, and financial support.

To our brothers, sisters, mentor, friends, and classmates who shared their words of advice and encouragements to finish this project.

ACKNOLEDGEMENT:

First, we want to thank our supervisor **Dr. Ahmed Yakoup** for his support, patience, kindness, unlimited help, and encouragement. He was always there for guiding us. He had done his best to support our project and our ideas, helped us more, and lifted our spirits.

Second, we want to thank **our Parents** for their patience and support. They tried so hard to be in our sides, encouraged us, and prayed for us.

Third, for **our faculty** for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Finally, we want to mention God's success that guides us to all of this.

Thank You All.

Table of Contents:

CHAPTER 1

INTRODUCTION

In this chapter, we will discuss the project's objectives, scope of the project, work needed to build the project, and overview of the whole document.

Problem Statement:

This application solves an important problem: How can the parents and their children find a good school with the specifications they want? Sometimes, the parents spend much time searching for a good school with special specifications, visit it, consume money and time then they discover that it wasn't the suitable school.
Our application facilitates searching for a school with the required specifications.

Project Objectives:

As we mentioned before, the main goal is to make it easy for the parents to get their school so:
The main objectives are:

- Both users and schools create an account on our application.
- The school can have a profile to show its information and instructions.
- The user can have a profile to show his data.
- There are some questions to the user which identify the special specifications that he wants, we called it a filtration page.
- Each school creates a quiz to test the students who want to join it.
- Each user can take one quiz to answer so the result will identify if he is qualified enough to join that school or not.
- The user can be a participant in a school in case he passes its quiz.
- Users can communicate with the schools by sending mails.

- The admin has the right to accept the school account, reject it or delete it.
- The user can choose a school as a favorite school and add it to the favorites to make it easy for him to follow its new updates.
- The user can search for a school.

Scope of work:

The approximate work involved to finish the project is divided into These five phases:

1. Planning:
   - Collecting data about the project and the lack that made us in a need of the website.
   - Making surveys.
   - Determining the functional and non-functional requirements.
   - Setting a Gantt chart for the project.
   - Determining the resources of the team.

2. Designing:
   - Determining the diagrams to be carried out within the project:
     - ✓ Database Schema.
     - ✓ Activity Diagram.
     - ✓ Class Diagram.
     - ✓ Use-case Diagram.
     - ✓ Sequence Diagram.
     - ✓ System Architecture.
     - ✓ Wireframe diagram.

3. Coding:
   - The main supposed functions to be coded in this App are:
     - ✓ Register as a user/school.
     - ✓ Login as a user/school.

- ✓ Show profile for user/school.
- ✓ Edit profile for user/school.
- ✓ Send E-mails.
- ✓ Create Quiz.
- ✓ Take Quiz.
- ✓ Show Quiz.
- ✓ Filtration.
- ✓ Search for school.
- ✓ Add to favorites.
- ✓ Remove from favorites.
- ✓ Show Favorite schools
- ✓ Rating.
- ✓ Accept/Reject School Account (Admin).
- ✓ Delete School (Admin).
- ✓ Show followers.
- ✓ Show Participants.
- ✓ Change Password.
- ✓ Forget Password.
- ✓ Logout.

4. Testing:
   - Functional Testing:
     - ✓ Unit Testing.
     - ✓ Regression Testing.
     - ✓ Integration Testing.
     - ✓ System Testing.
     - ✓ Acceptance Testing.
   - Non-functional Testing:
     - ✓ Performance Testing.
     - ✓ Stress Testing.
     - ✓ Load Testing.
     - ✓ Security testing.
   - Test Cases.

5. Documentation:
    - **Project Introduction:** includes an overview of the project and limitations.
    - **Project Background/Literature Review:** includes the tools and technologies as well as tasks and timeline plan.
    - **Project Methodology:** includes the functional and nonfunctional requirements, the diagrams, software tools, analysis of the project and the development approach.
    - **Project Testing:** includes testing types we used in our application.

Overview of Remainder of Document:

The remainder of the document will take us through the details of the project and will cover the following:

The general constrains and limitation of the project:

- Project benefits.
- Project's development approach.
- Software tolls.
- Planning and analysis of the idea of this project.
- Risk management.
- Project Requirements.
- Project' Software diagrams
- Project results.

CHAPTER 2

BACKGROUND/LITERATURE REVIEW

In this chapter includes citation of theoretical background, related work and general constraints should be included. We are going to discuss and go deeper in how we plan the project and the project benefits.

General Constraints:

- **Time:**

Task	Time
Searching for ideas and choosing one	1 month
Designing Diagrams	2 months
Searching for technologies and studying them	2 months
Time of each sprint	2 weeks
Implementing the whole project	5 months
Editing diagrams and Testing	2 weeks

- **Assign Tasks:**

  We assigned the tasks based on:
  - ✓ The team members physical ability.
  - ✓ The different skills of each member (Testing – Analysis – Programming).
  - ✓ The ability of searching for ideas, technologies, and techniques.
  - ✓ Knowledge and background about the project idea and the required techniques.
  - ✓ Agile planning with divided sprints.

- **Gathering data:**
  - ✓ The resources of the technology we want to use (Flutter).
  - ✓ A tutorial that explains how can we use a firebase in flutter?
  - ✓ How to build a system architecture diagram?

- **Manage behavior training:**
  - ✓ Limitations in understanding and getting the information and points quickly.
  - ✓ Limitations in selecting a meeting time appropriate to advisor.

Project Description:

- **Goal and Definition:**
  - ✓ This is a mobile application serves the parents, students, and schools.
  - ✓ It is an **Advanced Search** application which helps the parents and the children save their money and time, help them get a suitable, good school without wasting time going to a lot of schools to know what they provide and what they do.
  - ✓ The goal of this application is to prevent missing any year because of searching for a school and special specifications. You can find the school and your special specifications from home online.
  - ✓ The schools can save part of their marketing budget by creating an account on our application, showing their instructions and what they provide, and communicating with clients so they will market for themselves without spending much money.

- **Problems to be faced:**
  - ✓ If the framework stopped sharing an important feature, it'll be difficult to use it or replace it.
  - ✓ The time is tight.
  - ✓ Also, the budget we need to make every feature efficiently.
  - ✓ Anything would happen to any team member so the work's load will be on the rest of the team.

- **People and Organizations:**
  - ✓ Parents.
  - ✓ Students.
  - ✓ Schools.
  - ✓ Admins.

Assumptions and Constraints:

- **Assumptions for project:**
  - ✓ Know all about the technologies we will use.
  - ✓ The project will be delivered in an efficient time.
  - ✓ Know the basics of Firebase which makes us able to build the relations between all pages and features.
  - ✓ All the main features will be developed.
  - ✓ The whole project will work successfully.

- **Constraints for project:**

✓ Business constraints:
- The project should have specific categories.
- The project has only one admin.
- No money available to pay for paid resource.
- We have only one tester.
- We use agile planning.

✓ Technical Constraints:
- We have some missing features that will be developed in the future.
- Github.
- All schools must be approved by the admin.

Context and Background:

- **Context:**

✓ Political:

Our project will be free, don't have to pay for. It is all about searching for a school in Egypt with the same specifications as yours.

✓ Sociological:

Our project is a user-friendly application any one can use it.
The following sectors can use it easily: Users (Parents/Student), Schools, and admins.

✓ Legal:

We used agile planning so we divide the whole features to separated sprints. Each sprint will take 2 weeks and contains a small task for each team member.

✓ Environment Factors:

We don't have any environment factors but the application shows the result of searching based on the schools which have account. Clearly, if a user enters a data about special specifications that he wants and these specifications are the same as a school that doesn't have an account so the application will not show it as a result.
That if we consider this as environment factors.

- **Background:**

✓ Expected Value:

We expect to reduce money, save time, make it easy for any student to find his/her school with his/her special specifications, search for any school in any time, reach your destination on the certain time, reduce efforts, and communicate with the schools easily and rate them.

✓ Pre-requisite knowledge:

- Flutter as a Framework.
- Dart as a programming language.
- Firebase.
- Agile planning.

We divided the team into two categories: Frontend – Backend
4 team members for Frontend and 2 team members for firebase.

Project Benefits:

- Finding a school that has all the same constraints that the user wants on time without missing an educational year.
- The user can communicate with the school at any time and without wasting money.
- The user can see all the school updates from home.
- The school will get more students easily.

Beside:

- Studying Flutter.
- Studying Dart.
- Know how to apply agile.
- Know basics about Firebase.

CHAPTER 3

METHODOLOGIES

Development Approach and Methodology:

- **Approach:**

Along our academic years, we have studied the approaches of developing a software. We have decided to set the iterative approach for our project for some reasons that will be discussed.

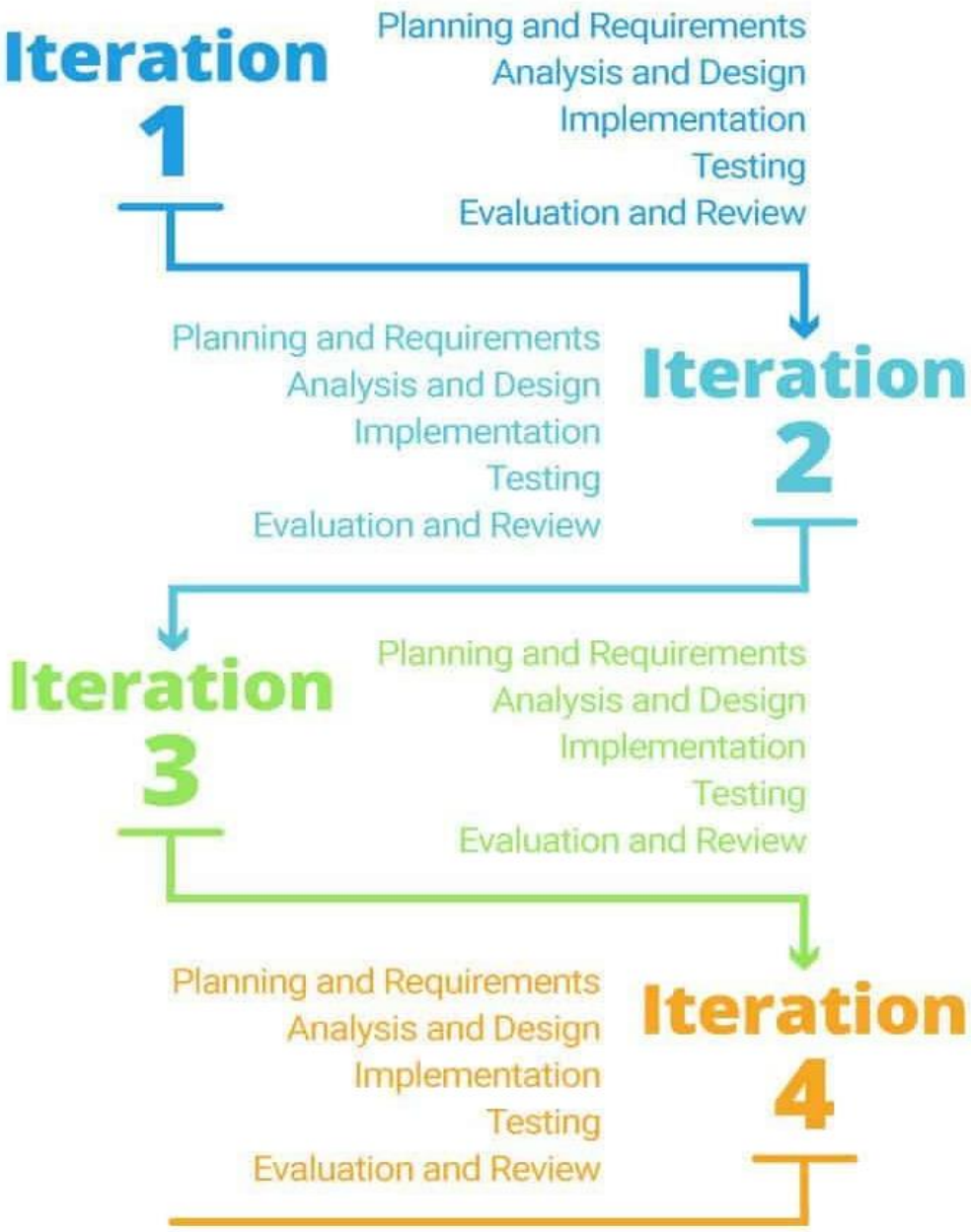
Iterative Approach:

Iterative development is a way of breaking down the software development of a large application into smaller chunks.

In iterative development, feature code is designed, developed and tested in repeated cycles. With each iteration, additional features can be designed, developed and tested until there is a fully functional software application.

- ✓ The purpose of working iteratively is to allow more flexibility for changes.
- ✓ By working iteratively, the project team goes through a cycle where they evaluate with each iteration, and determine what changes are needed to produce a satisfactory end product.
- ✓ The process relies on a continual cycle of planning, analysis, implementation, testing, and evaluation. Starting from the initial planning.
- ✓ Then the first, prioritized portion of the project becomes the initial cycle of development.

This figure explains how the iterative approach works:



Methodology:

To cope up with incremental approach we have selected a methodology would help us in developing software.

We selected agile Development methodology.

Agile Development Methodology:

It is a methodology of developing a software and contains some advantages, so we selected this methodology.

- **Pros:**
  - ✓ It minimizes risks such as bugs, cost overruns, and changing requirements.
  - ✓ It allows software to be released in iterations.
  - ✓ Iterative releases improve efficiency by allowing teams to find and fix defects and align expectation early on.
  - ✓ It also allows users to realize software benefits earlier, with frequent incremental improvements.
  - ✓ It makes clients involved in the process to get feedbacks from them.

This figure represents the idea of agile:

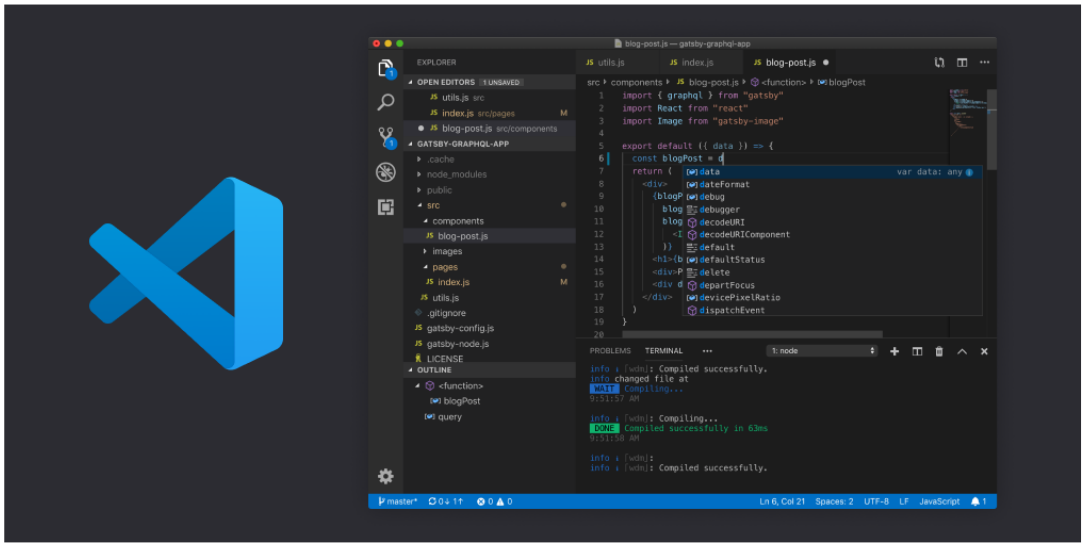
There are a lot of types inside agile, but we selected Scrum methodology and it consists of:

- ✓ Know all features and putting them in backlog.
- ✓ Divide the project into number of sprints.
- ✓ Dividing each sprint into number of user stories.
- ✓ Each sprint contains three columns To Do, Doing, Done.
- ✓ Put tasks in each Sprint and then in each column.
- ✓ Assign each task to one team member or to a group of them.
- ✓ Set a deadline for each sprint.
- ✓ Set a deadline for each task.
- ✓ Choose a repository to upload tasks on.

Hardware and Software Tools:

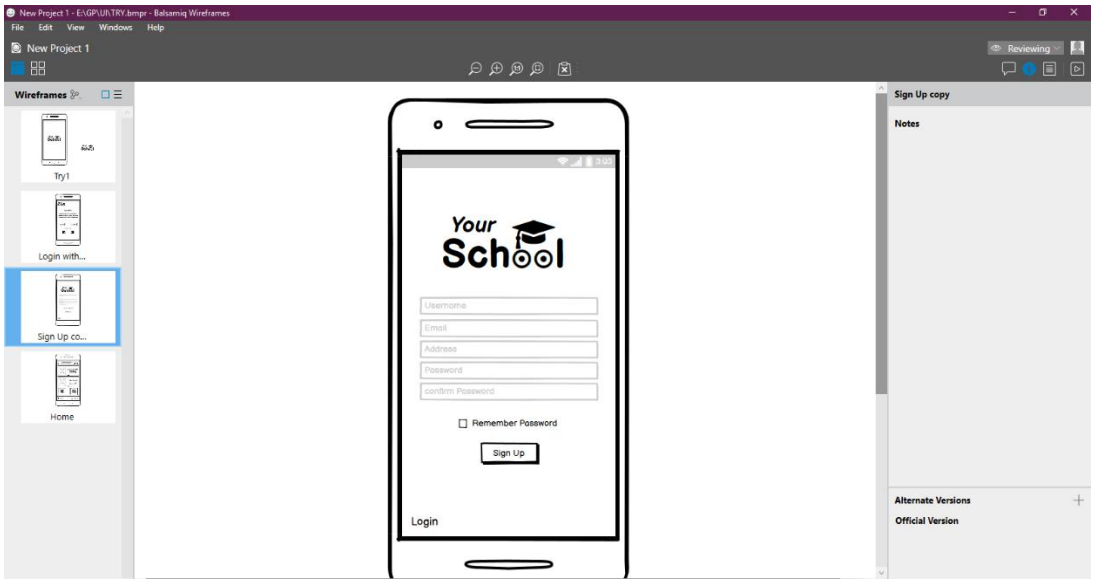
- **Software Tools:**
  - ✓ Android Studio: editor

  - ✓ Visual Studio Code: editor

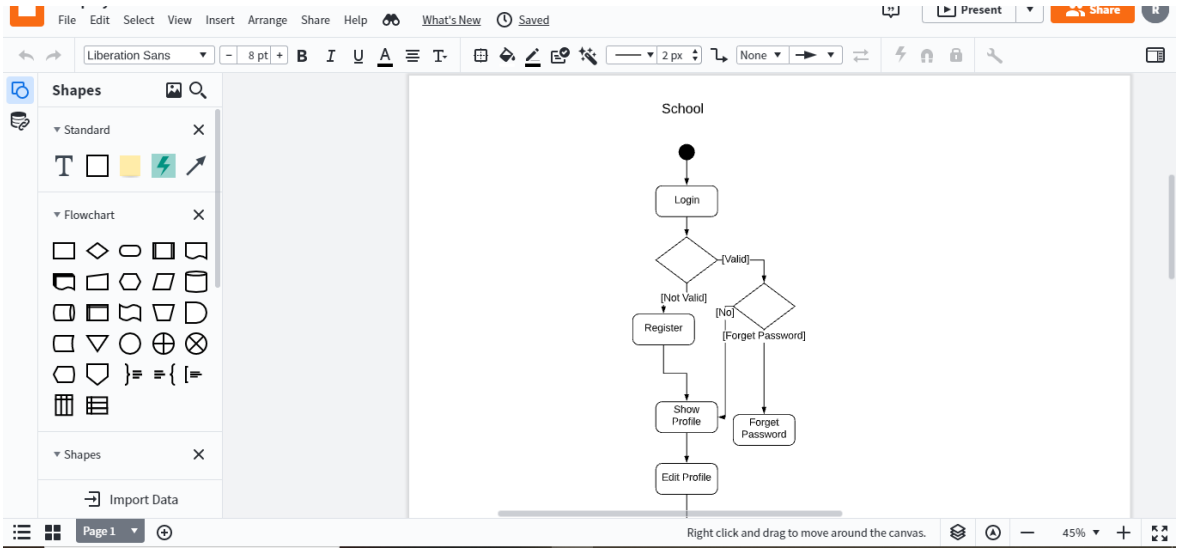


- ✓ Firebase : local database

- ✓ Balsamic Wireframes: for wireframes and UI:



- ✓ LucidChart: For designing diagrams



- **Hardware Tools:**
  The project doesn't require any hardware tools.

Planning And Analysis:

- **Project planning:**
  - ✓ Feasibility Study:
    A feasibility study used to determine if the project is achievable, so we will show some points to determine it.

    Problems that the project solves:
    - The difficulties to reach a suitable school with the required preferences.
    - The problem of missing the year without getting a suitable school.
    - The difficulties for the school to make a lot of interviews for all applicants.
    - The school pays much money for marketing to itself.

Who will benefit from the project?

- Students.
- Parents.
- Schools.

Why we chose this problem to work on:

- Help some people get a school with their preferences.
- Help schools facilitate their interaction with a lot of applicants in the filtration process.
- Help users don't miss the year.
- Save time for users and schools.
- Help schools in marketing.
- Help parents to know everything and all instructions about the school.

Market analysis:

- **Main Competitors:**
  - ✓ Easy Schools
  - ✓ Nearby Schools Finder
  - ✓ Nearest Places
  - ✓ دليل المدارس

- **Unique points in our project:**
  - ✓ Using our application will save a lot of time for schools and users.
  - ✓ Schools can create a quiz for users to facilitate the upcoming processes.
  - ✓ Users can take the quiz, contact the school and know everything about the school without wasting their time.
  - ✓ Using our application facilitates marketing to the schools
  - ✓ The application is organized.
  - ✓ The UI is friendly with suitable colors.
- **Target Market:**
  - ✓ Age: 6– 19.
  - ✓ Gender: Both.

- ✓ Place: anywhere inside Egypt.
- ✓ Education level: Primary, Preparatory, and Secondary.
- ✓ Relevant behavioral factor: can use a smart devices and dealing online with different languages.

Organizational and operational analysis:

- Number of team members this project will need:
  - ✓ 5 developers.
  - ✓ 1 tester.
  - ✓ 1 designer.
  - ✓ 1 admin.
  - ✓ 1 Public relation executives.

- To keep the work on this project we shall provide:
  - ✓ Rules Manual.
  - ✓ Regular Trainings.
  - ✓ Safe Environment.
  - ✓ Modern Management.
  - ✓ Good workplace.
  - ✓ Efficient work tools.

Sales Strategy:

To increase sales we have to take care of important points:

- ✓ Check for and follow up users' complaints regularly.
- ✓ Execute regular testing.
- ✓ Upgrade the application to offer the users optimized functions.
- ✓ Claim and Maintain high rate on the application.
- ✓ The application is completely free.

Technical analysis:

No Hardware required for the app.

Main technologies and programs we're associated with:

- ✓ Flutter Framework.
- ✓ Firebase Storage.
- ✓ Microsoft Visual studio code.
- ✓ Balsamic wireframes.
- ✓ Lucidchart.

Resource and Time analysis:

- **Resources needed:**
  - ✓ Devices for coding, running and testing – Laptops and mobile phones.
  - ✓ Coding programs.
  - ✓ Internet service.
  - ✓ Design programs.
  - ✓ Flutter libraries and packages.
- **Time schedule:**
  - ✓ 1 month for planning.
  - ✓ 2 months for training.
  - ✓ 6 months for developing and testing.

Estimated Cost:

The budget of the application as programming and its process includes:

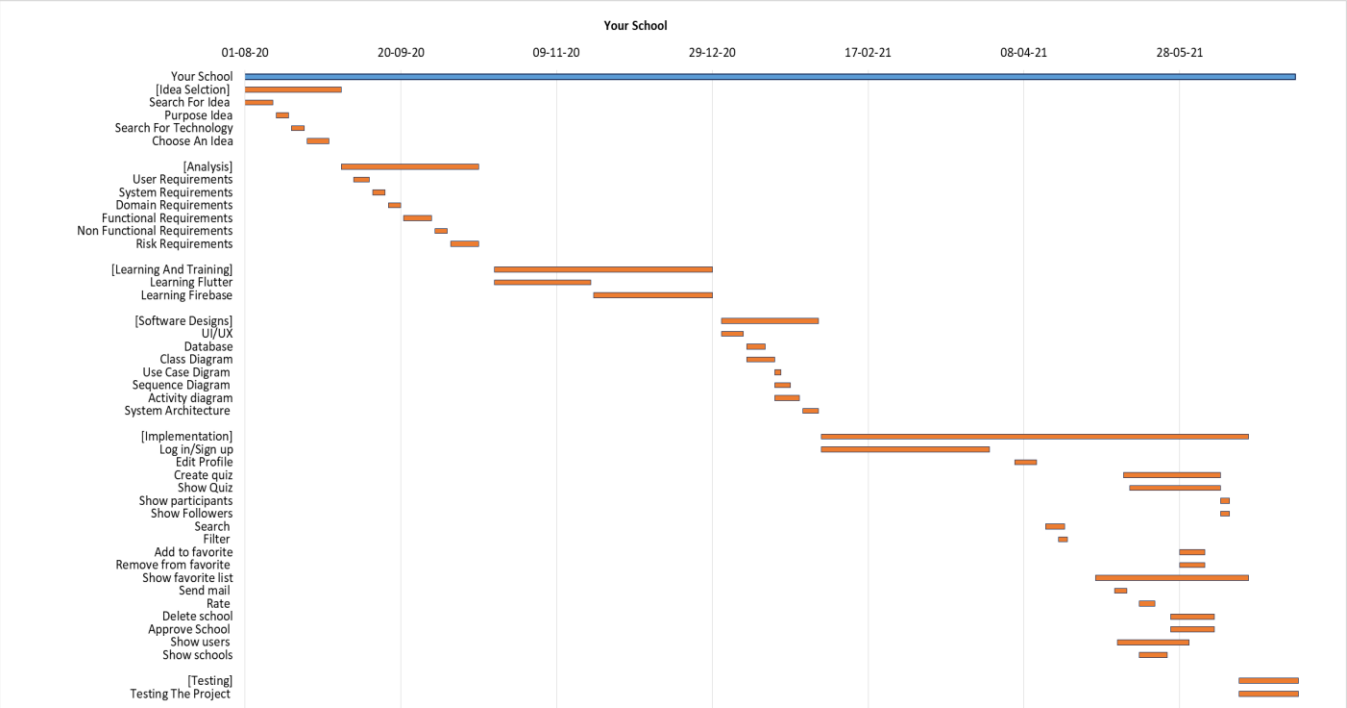
Purchasing a flutter course = 200$.

Marketing for the application = 99$\year.

Meeting in a public workspace.

Offline meeting every week (for 2-3 hours) in public workspace (100 LE).

Gantt Chart:



Analysis and Limitation of Existing System:

- **Assumption and Constrains:**

  We have existing competitors but we will discuss the features and what makes our application unique.

  At first this table defines the features they have or not:

At first this table defines the features they have or not:

App Logo	App Name	Filter & Search	School Profile	User Profile	Take Quiz
	Easy Schools	T	T	T	F
	Nearby Schools Finder	F	F	F	F
	دليل المدارس	T	T	F	F
	Nearest Places	F	F	F	F
Your School	Your School	T	T	T	T

Now we will discuss each one individually:

- **Easy Schools:**
  - ✓ Description:
    It is the biggest schools platform in Egypt.
  - ✓ Features:
    - Search for schools by name & via filtration.
    - Show schools and user's profile.
    - Show favorite list for schools.
    - Compare between schools.
- **Nearby Schools Finder:**
  - ✓ Description:
    This application shows list of schools in your nearby area with more detail.
  - ✓ Features:
    There are no features in this app except getting the nearest school to your location.
- **دليل المدارس:**
  - ✓ Description:
    Application of the Egyptian schools guide, expenses and coordination of schools.
  - ✓ Features:
    - The application can search for schools with filtering by governorate, type of school and administration.
    - The application can show the school's profile.
    - The user can communicate with the school.
- **Nearest Places:**
  - ✓ Description:
    This application allows the user to choose the nearest school, hotel, airport, etc.

- ✓ Features:
  - It has different places to search for including schools.
  - It shows the location on the map.

Risk and Risk Management

	Risk	Probability rate	Impact rate
Schedule Risk	Wrong time estimation.	Low likely	High
	Resources like team and their skills are not tracked properly.	Mid likely	High
	Failure to identify time required to develop functions.	Mid Likely	Mid
	Unexpected project growth.	Most likely	High

Table 3-schedule risk

Can be avoided by:

- Following up the Gantt chart regularly and reduce the critical paths in the project network.
- Look after the major tasks -specially- that the other tasks depend on.
- Monitor\review the agreed-on time plan. o Scheduling the risky tasks first.

Can be reduced (if happened) by:

✓ Updating the plan to adapt with any new added requirements without affecting the already existed plan.
✓ Maximize the rest of resources like team and tools. o Not wasting time because the problem but move forward.

	Risk	Probability rate	Impact rate
Budget Risk	Wrong budget estimation.	Most likely	Low
	Cost overruns	Low likely	Low
	Project gr owth	Low Likely	Mid

Table 4 - budget risk

Can be avoided by:

- Good research.
- Regularly checking the budget spent and needed.
- Never skipping the budget through any new added requirements.

Can be reduced (if happened) by:

✓ Sacrifice not-so-needed resource that consume money. o Quick adaptation.
✓ Replace cost-consuming parties with other available-resource consuming parties.

	Risk	Probability rate	Impact rate
Operational Risk	No resource planning.	Low likely	Mid
	No communication in the team.	Low likely	Mid
	No distribution for responsibilities.	Mid Likely	High
	Not enough training.	Mid likely	High

Table 5 - operational risk

Can be avoided by:

- Monitoring and evaluations at regular intervals. o Cutting the project into very small tasks.
- Full training before proceeding the work.

Can be reduced (if happened) by:

✓ Adding pieces of training into the normal work.
✓ Rotation through the training.
✓ Changing the responsibilities to achieve the best optimized output.

	Risk	Probability rate	Impact rate
Technical Risk	Changing requirements.	Most likely	Mid
	Angular and node js are in their initial stages.	Most Likely	High
	The database is not efficient.	Low likely	Mid
	Attacks or application f ailure	Mid likely	High

Table 6 - technical risk

Can be avoided by:

- Preparing the threats and actions resulted.
- Understanding the platform\tools used.
- Not rushing the analysis phase.
- Specifying the desired outcome.
- Detailed test cases and regulations.

Can be reduced (if happened) by:

✓ Analyzing the security threats.
✓ Preparation for facing any attack.
✓ Quick adaptation to any new requirements.
✓ Evaluating \monitoring the database from time to time.

Requirements:

- **User requirements**
  - ✓ **Mandatory Requirements:**
    - The user can make an account on the application by entering his data easily.
    - The user can search for specific school.
    - The user can search for a desired school through a filtration process.
    - The school also can make an account easily.
    - The admin can approve or cancel the school's request.
    - The admin can delete a school and show all registered users & schools.
    - The school can make its own quiz and show it.
    - The user can take the quiz.
  - ✓ **Desirable Requirements**
    - The user can add a school to favorite or remove it.
    - The user can show a list of favorite schools.
    - The user can communicate with the school via mail.
    - The user can rate the school.
    - The school can show the users following it.
    - The school can show its participants.

System requirements:

Application:

Since our application will be made by flutter, then it will work on android and IOS phones. It can also work on browser through (Flutter Web).

Internet Connection:

Since all schools should be approved first from the admin, the admin can do that via a browser, so we need a good internet connection and also we need that to keep data on cloud firestore.

Storage space

In order to keep the users and schools' data, we need a fine storage.

Domain Requirements:

- Multiple users must be able to use the application simultaneously.
- The necessary software required to run the application.
- A server must be set up to host the database.
- The database should be backed up every once in a while in case the original does become corrupt.
- The application must have update capabilities for future models.

Functional Requirement:

Register

Actor: All Users (includes users and schools)
Pre: download the application and enter the data then click the "Register" Button
Description: This function is to make the user register his account with email and password to be able to use the application.
Post:
To user:
The home with many schools will appear to the user.
To school:
The features of the school are viewed in its home page.

Login

Actor: All Users and the admin.
Pre: the user must be registered on the application and click the "Login" button when opening the app.
Description: This function is to make the user log in to his account with password and email.

Post: the user will successfully enter his account and open the app.

Forget Password

Actor: All Users (includes users and schools)
Pre: the user must be registered on the application and click the "Forget Password" button then enter his email.
Description: This function is to enable the user to register with a new password if he isn't able to remember the old one.
Post: the user password is edited successfully.

Edit Profile

Actor: All Users (includes users and schools)
Pre: the user should open the application and click the "Edit Profile" button.
Description: This function enables the user to update his information.
Post: the user profile is edited successfully.

Change Password

Actor: All Users (includes users and schools)
Pre: the user should login the system.
Description: This function is to enable the user to update his password if he remembers it.
Post: the user password is updated successfully.

Show Profile

Actor: All Users (includes users and schools)
Pre: the user must log in at first and click on his profile.
Description: this function enables the user to view his profile with all his information.
Post: the user profile is viewed successfully.

Filter

Actor: users
Pre: the user must log in and open “Filtration” page then enter the data in the fields and click on “Filter” button.
Description: this function allows the user to search for a specific school with his preferences.
Post: A list of some schools appears to the user as he needs.

Search

Actor: users
Pre: the user must login firstly
Description: this function enables the user to search for a specific school by its name.
Post: the user gets the school successfully.

Create Quiz

Actor: Schools only
Pre: the school must log in firstly.
Description: this function enables the school to make its own quiz for the students.
Post: the quiz is created successfully.

Take Quiz

Actor: users only
Pre: the user must log in firstly.
Description: this function enables the user to have the quiz one time and answer it to get the result
Post: the user takes the quiz successfully, get his result and the quiz is closed to the same user.

Show Participants

Actor: Schools only.
Pre: the user should have taken the quiz and the school must login firstly to show the participants.
Description: this function enables the school to view users who pass the quiz.
Post: List of participants viewed successfully.

Add school to favorite

Actor: users only.
Pre: the user must log in firstly.
Description: this function enables the user to save the school to be able to view it next time easily.
Post: school is added to favorite successfully.

Show favorite schools

Actor: users only.
Pre: the user must log in firstly and add schools to favorite.
Description: this function enables the user to show the schools he added to favorite.
Post: list of favorite schools is viewed successfully.

Rate

Actor: users only.
Pre: the user should log in firstly and view the school's profile.
Description: this function allows the user to give rate to the school from 1 to 5.
Post: the school is rated successfully.

Send mail

Actor: users only.
Pre: the user should log in firstly and view the school's profile.
Description: this function enables the user to contact the school via mail.
Post: mail is written and send successfully.

Show followers

Actors: schools only.
Pre: the school should log in firstly and the user must add a school to favorite.
Description: this function helps the school view users who added this school to favorite.
Post: list of users is viewed successfully.

Approve request

Actor: Admin only.
Pre: admin must log in firstly and the school must register.
Description: this function enables the admin to accept the school to be registered.
Post: school is approved and registration is done successfully.

Cancel Request

Actor: Admin only.
Pre: admin must log in firstly and the school must register.
Description: this function enables the admin to cancel the school's request to be registered.
Post: the request is canceled successfully.

Show users

Actor: admin only.
Pre: the admin must log in firstly and users is logged in.
Description: this function allows the admin to view all registered users.
Post: a list of users is viewed successfully.

Show schools

Actor: admin only.
Pre: admin must log in firstly and schools is registered.
Description: this function allows the admin to view all registered schools.
Post: a list of registered schools is viewed successfully.

Delete school

Actor: admin only.
Pre: the admin must log in firstly and the school must be registered.
Description: this function allows the admin to delete a school from the system.
Post: school is deleted successfully.

Show quiz

Actor: schools only.
Pre: the school must log in firstly.
Description: this function allows the school to view its quiz.
Post: quiz is viewed successfully.

Non-Functional Requirements

- **Availability:**
  The application will be available to anyone and it can be accessed 24 hours in 7 days.
  The storage in database is very huge.
- **Efficiency:**
  Our application utilizes scarce resources: CPU cycles, disk space, memory, bandwidth, etc.
- **Flexibility:**
  The users and school's data can be updated or deleted.
  The user can take the school's quiz easily.
  All the main tasks are shown in the dashboard with different colors, all the buttons are obvious.

- **Reliability:**

  Our application is capable of the software to maintain its performance over time.

  Unreliable software fails frequently, and certain tasks are more sensitive to failure.

- **Scalability:**

  The application will be updated so it can serve more users (that increases gradually), process more data and do more functions.

  Any added feature or function mustn't affect the storage massively.

  Time to save any data mustn't increase with users' number increasing.

  Functions and roles shall be scaled for the number-growth of the users.

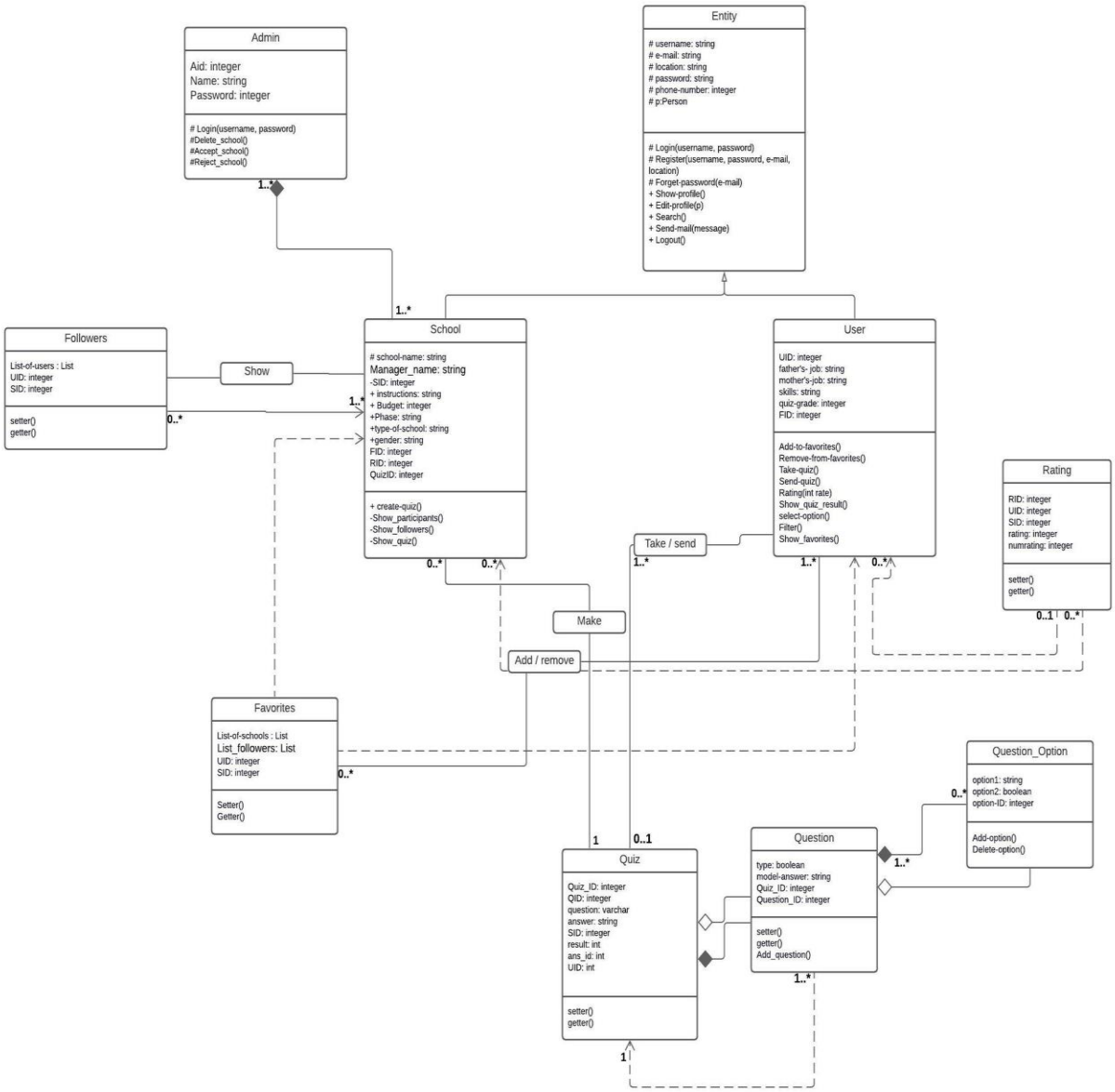
  Our application is scalable has the ability to handle a wide variety of system configuration sizes.

  The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines, etc.).

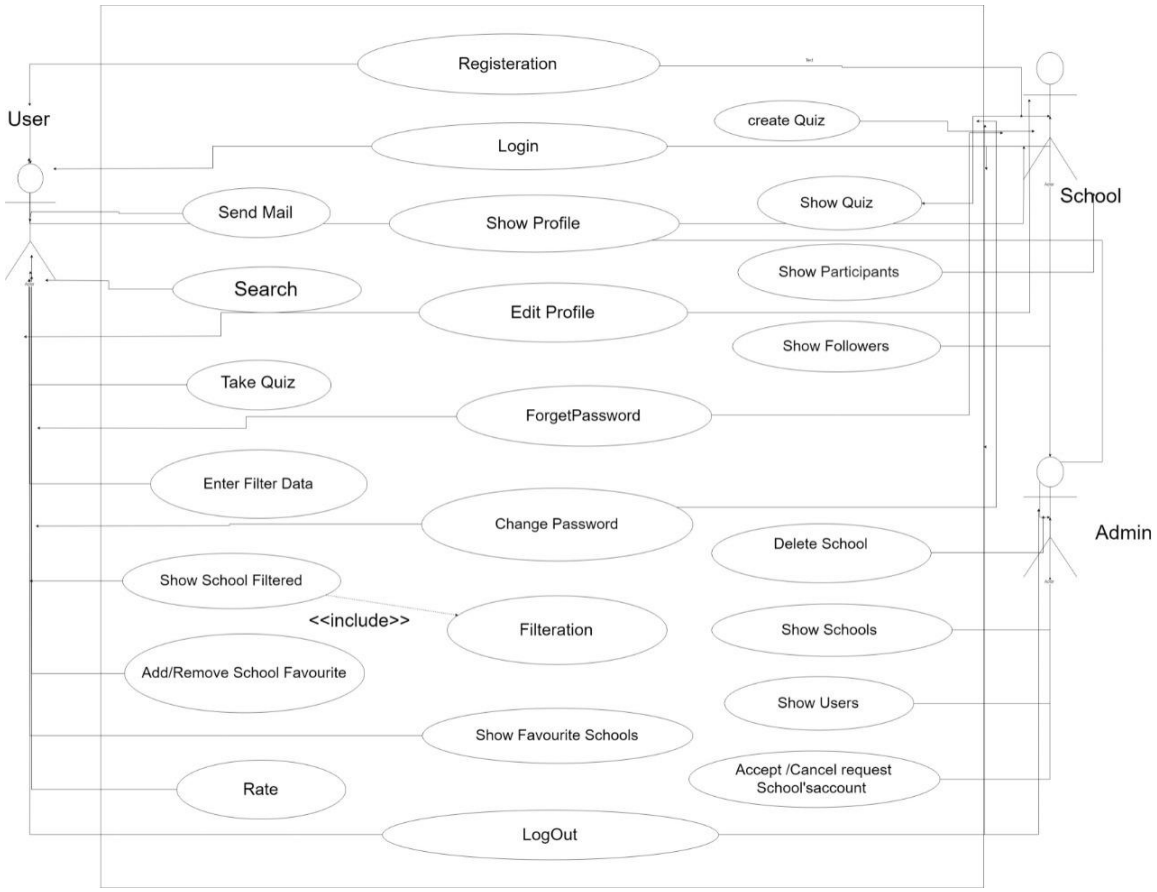
- **Usability:**
  Each school profile contains all its details to be obvious and easy for users to understand and know everything about the school.
  Both users and schools can add their data within few clicks, which will take no time to submit.
  The accessibility of the interface is very important since any type of people are likely to use the app.
  The application shall be easy to use by adult members.

Software Design:

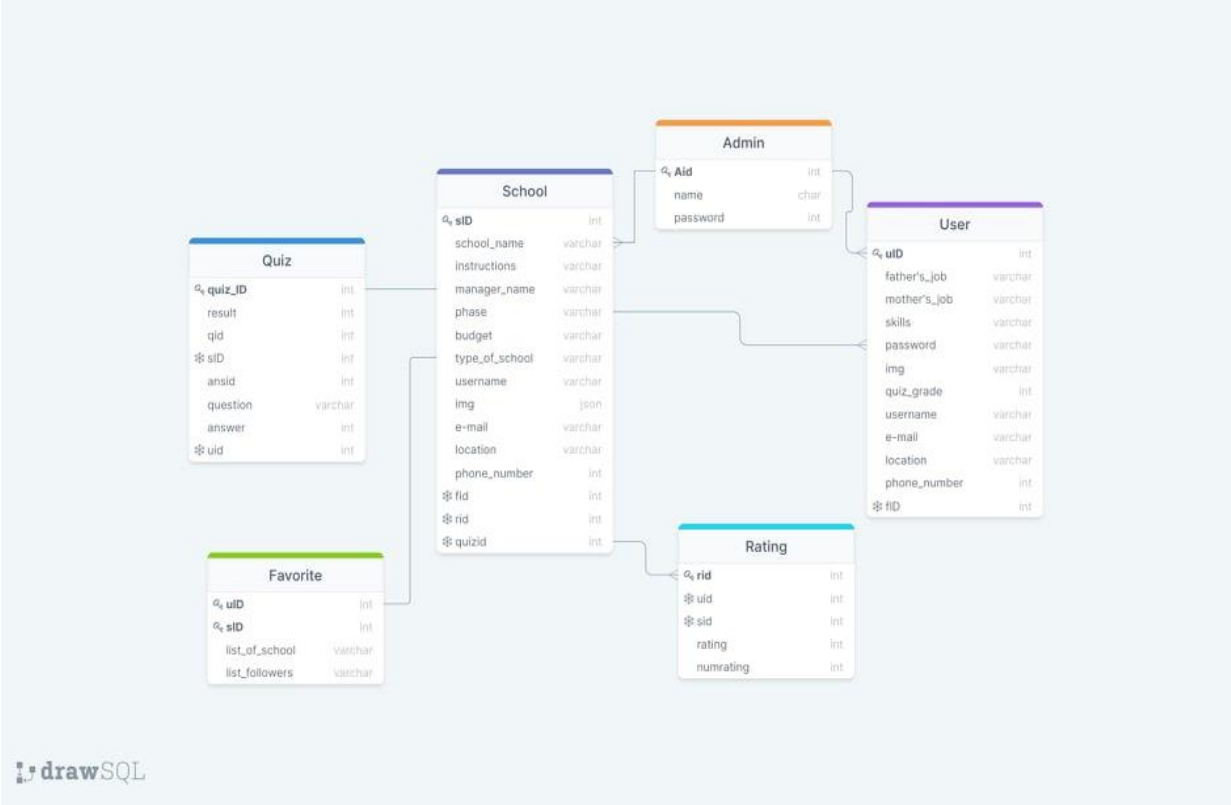
- **Class diagram:**



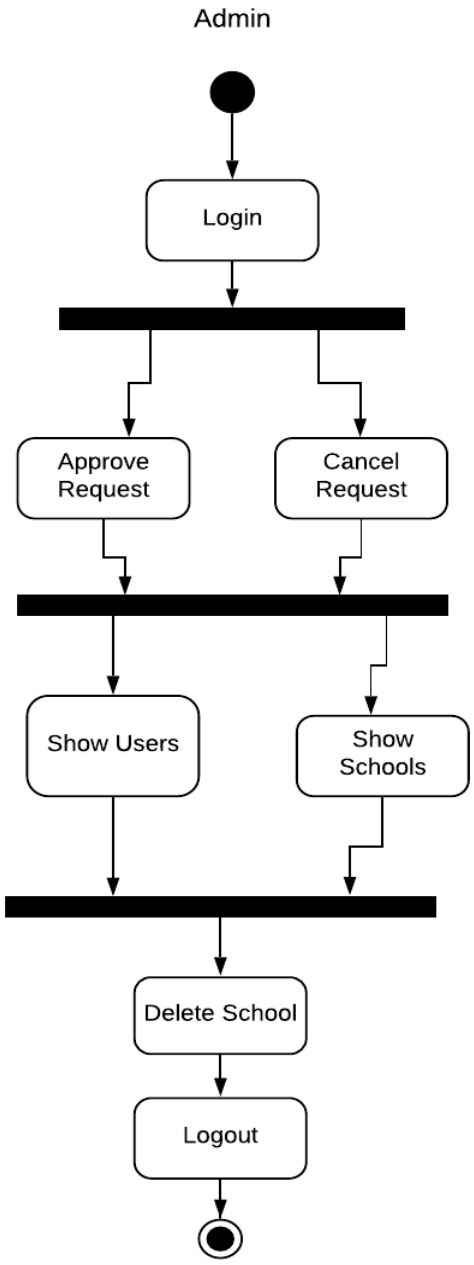
- **Use case diagram:**

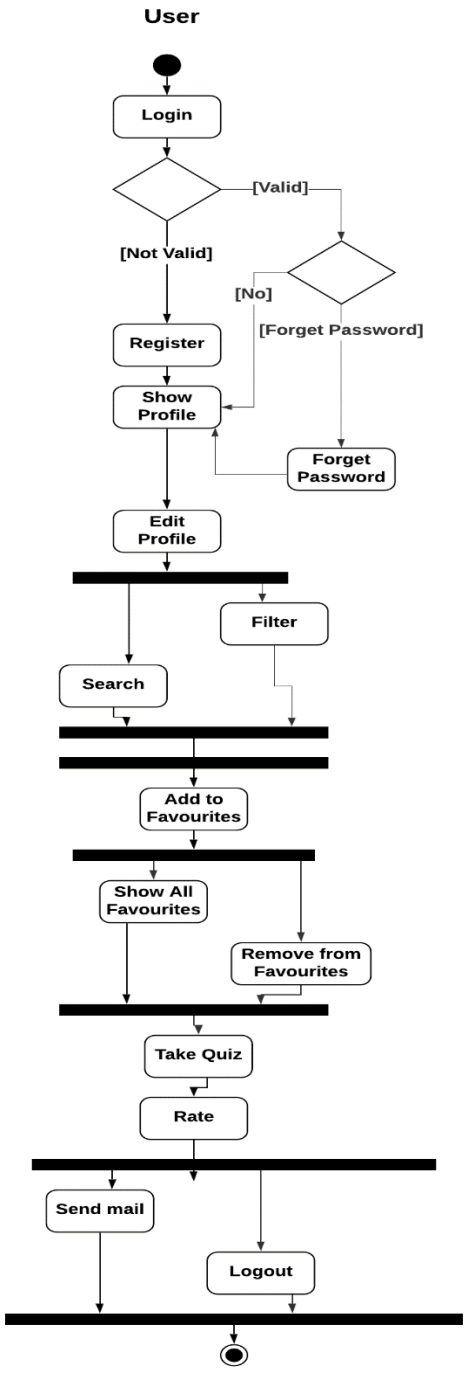


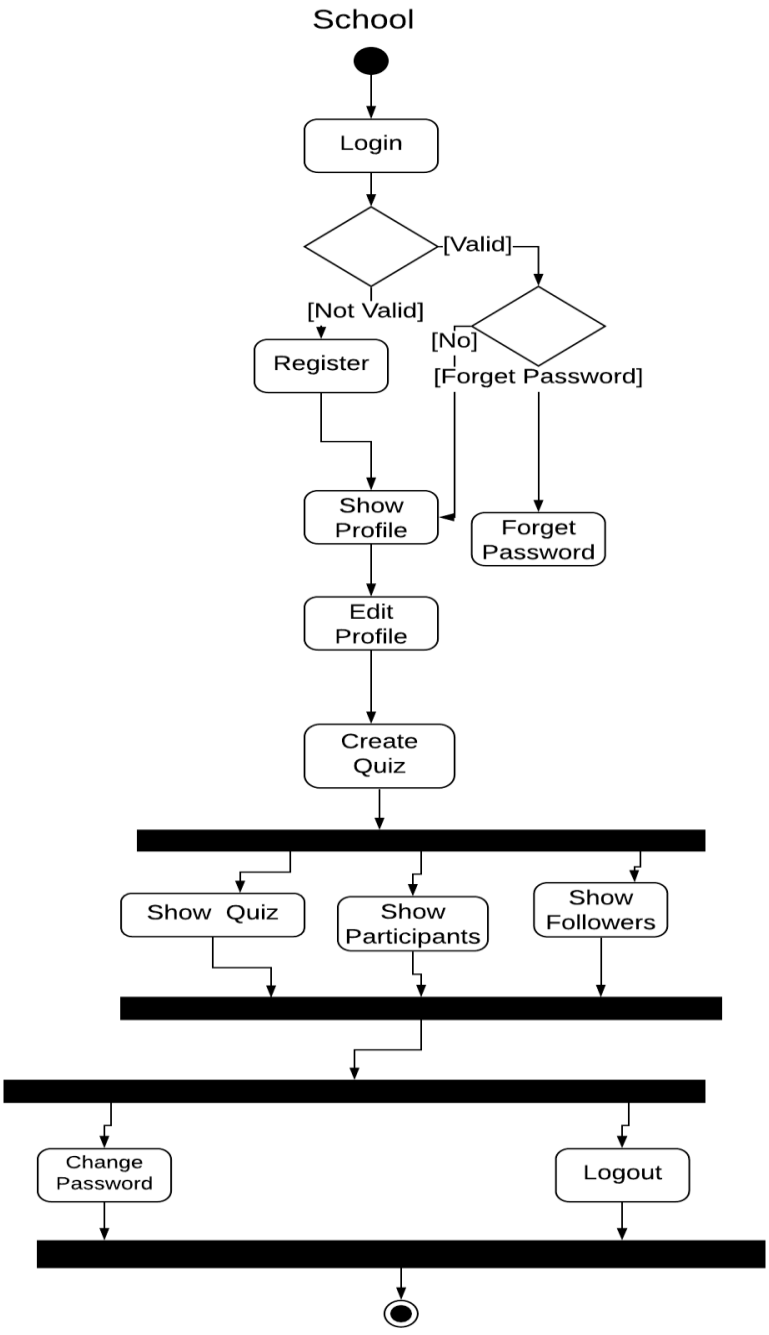
- **Database Schema:**



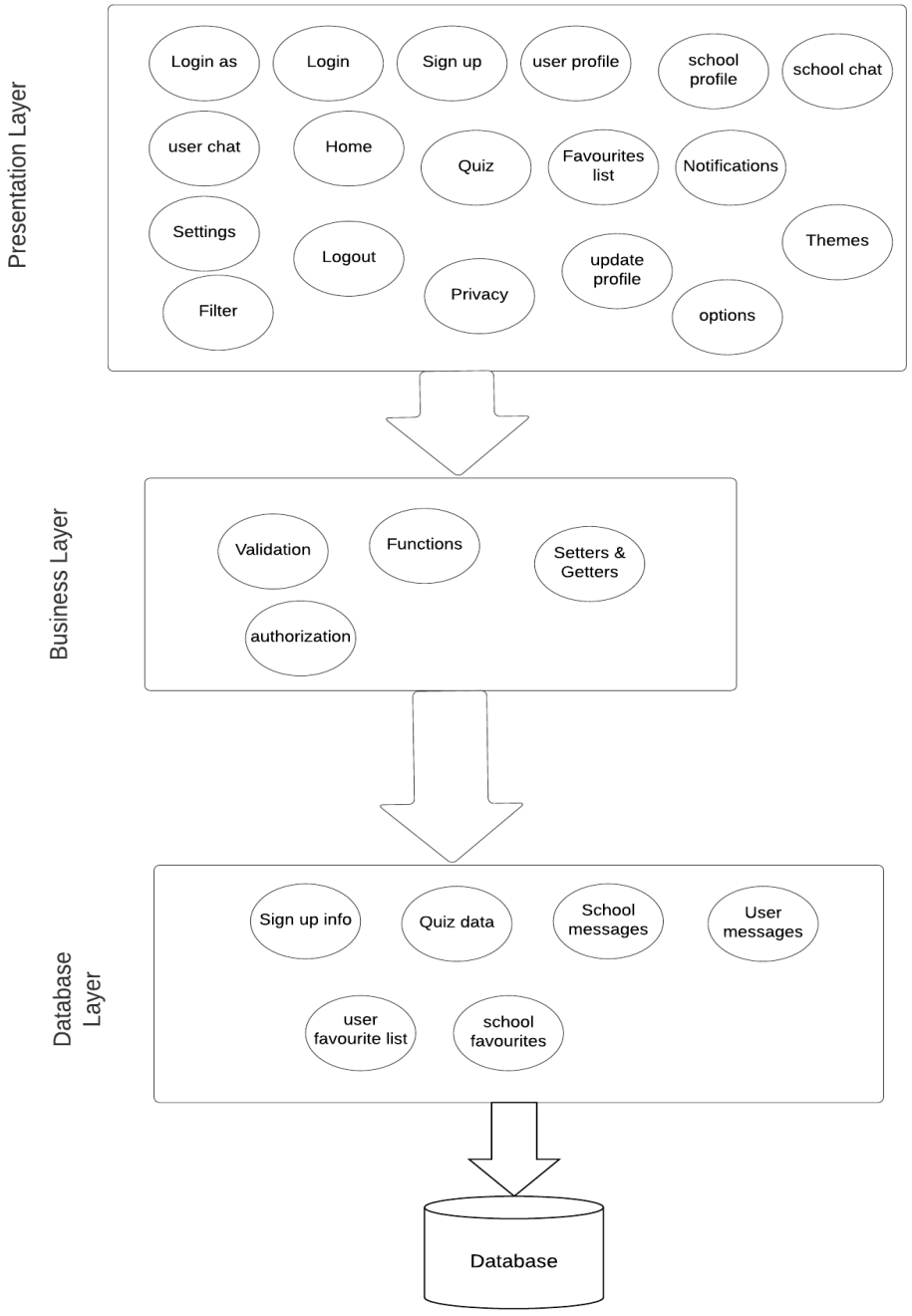
- **Activity Diagram:**





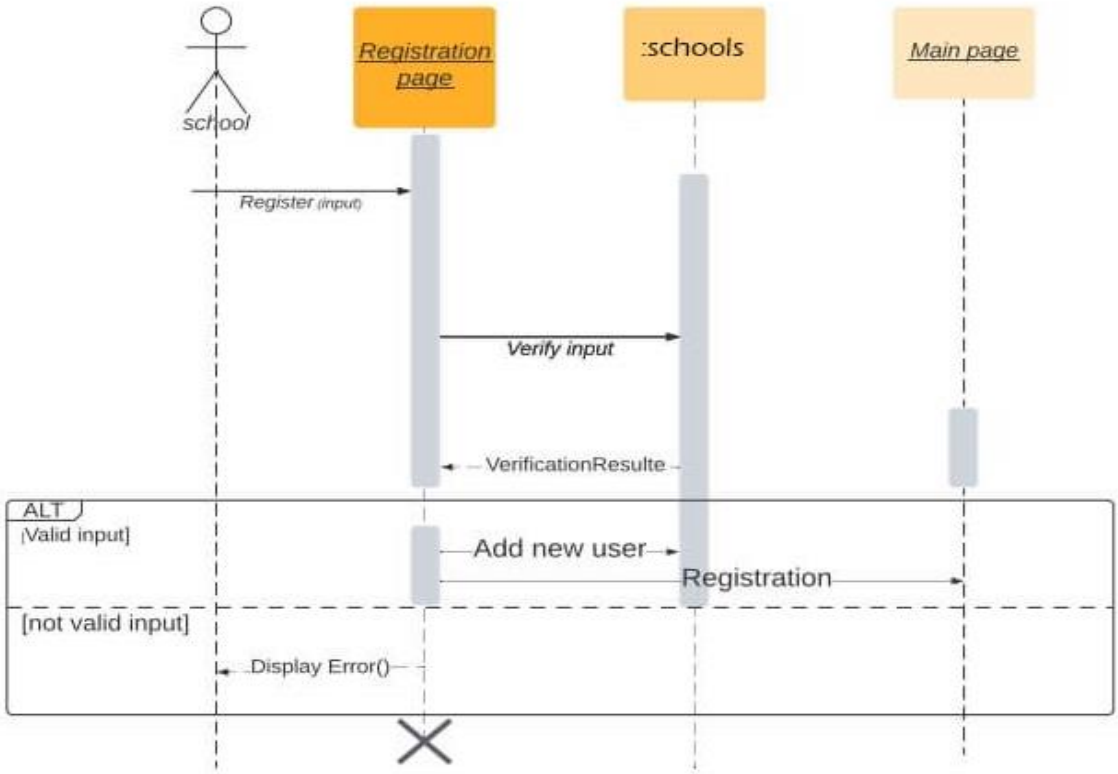

School
Login
[Valid]
[Not Valid]
[No]
Register
[Forget Password]
Show Profile
Forget Password
Edit Profile
Create Quiz
Show Quiz
Show Participants
Show Followers
Change Password
Logout

- **System Architecture:**

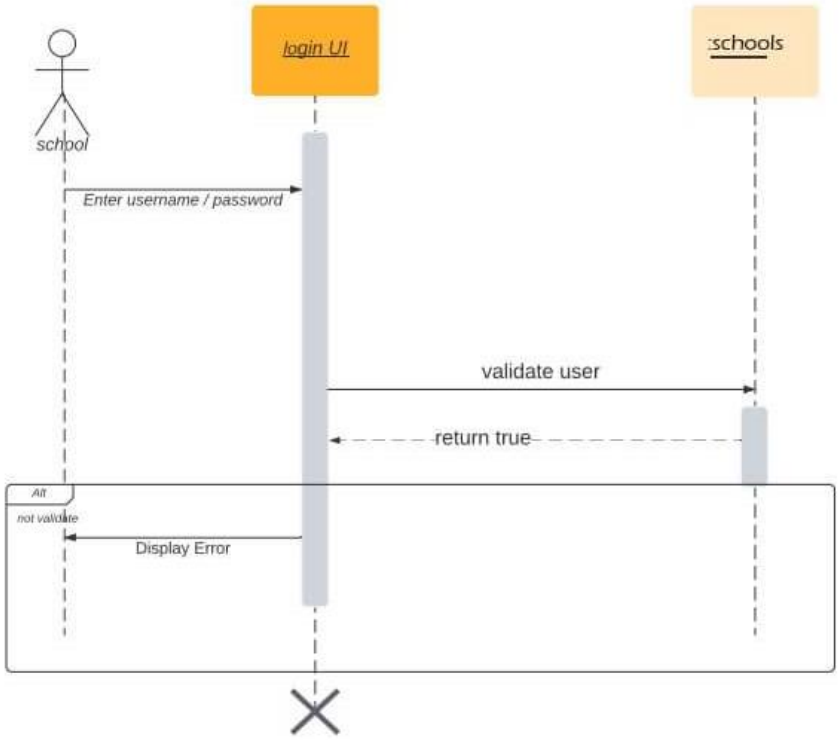


- **Sequence Diagrams:**

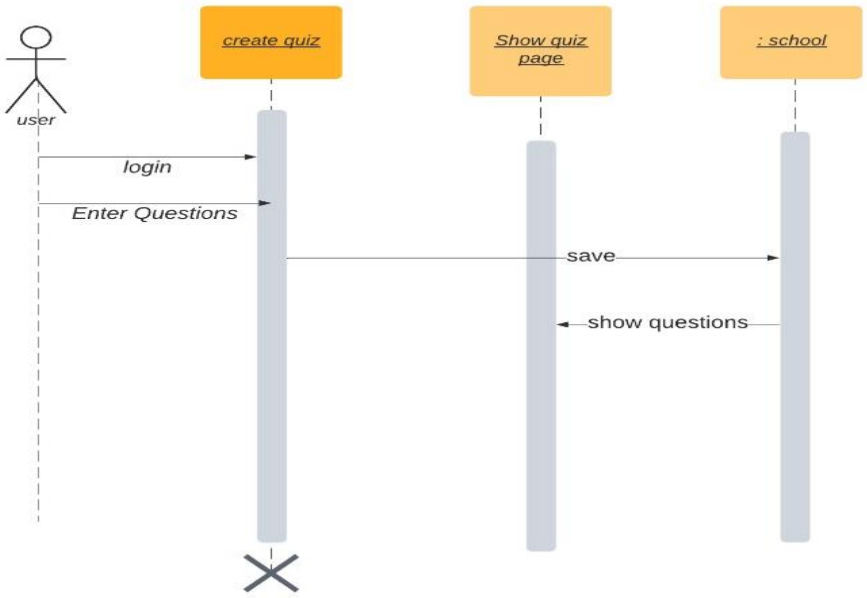
School Register:



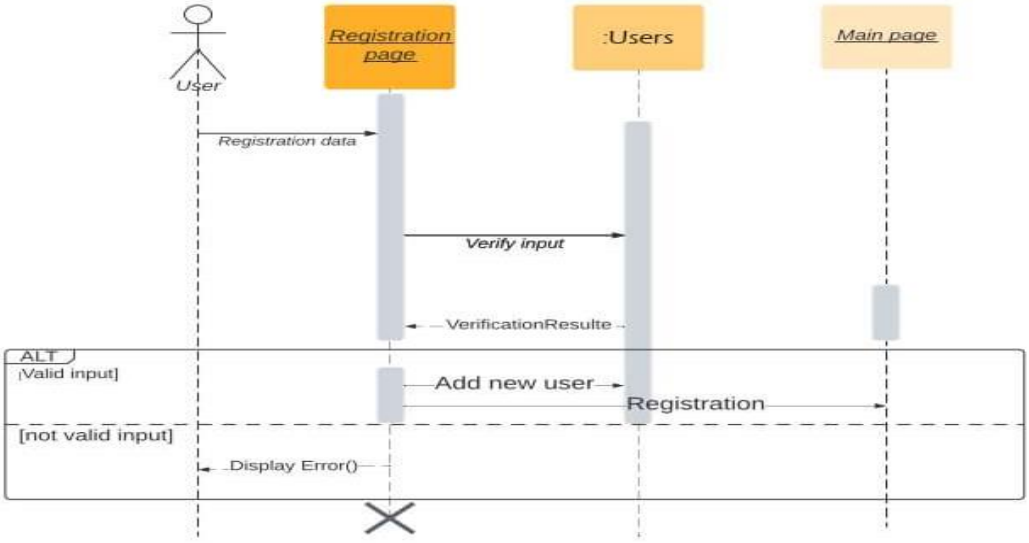
School login:



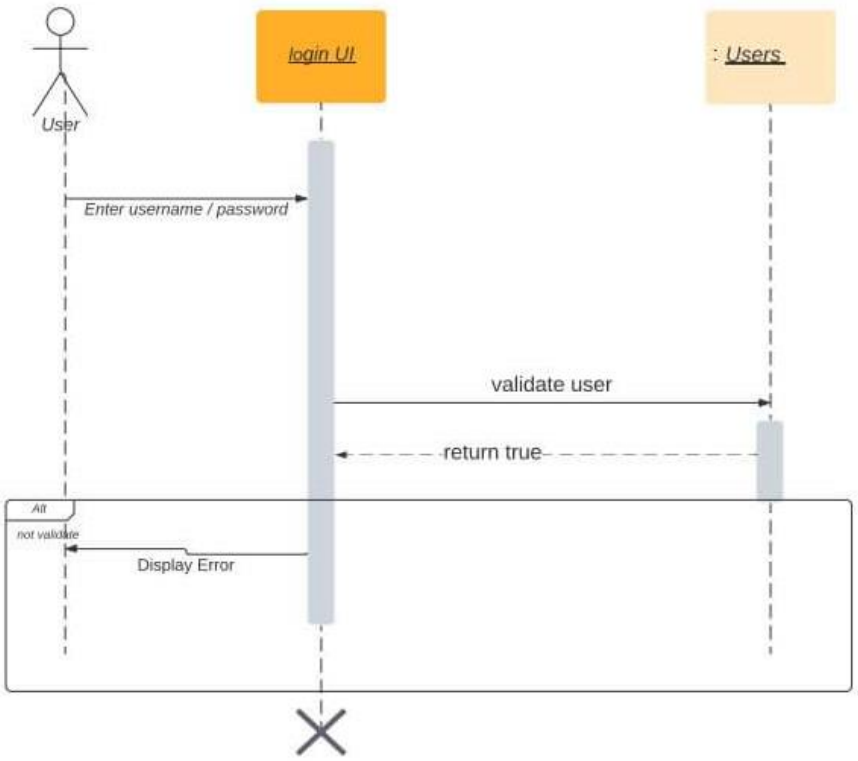
Create Quiz:



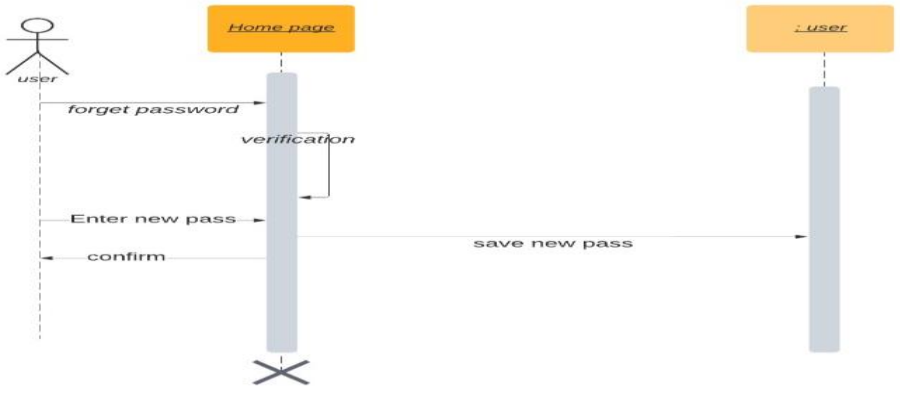
User Register



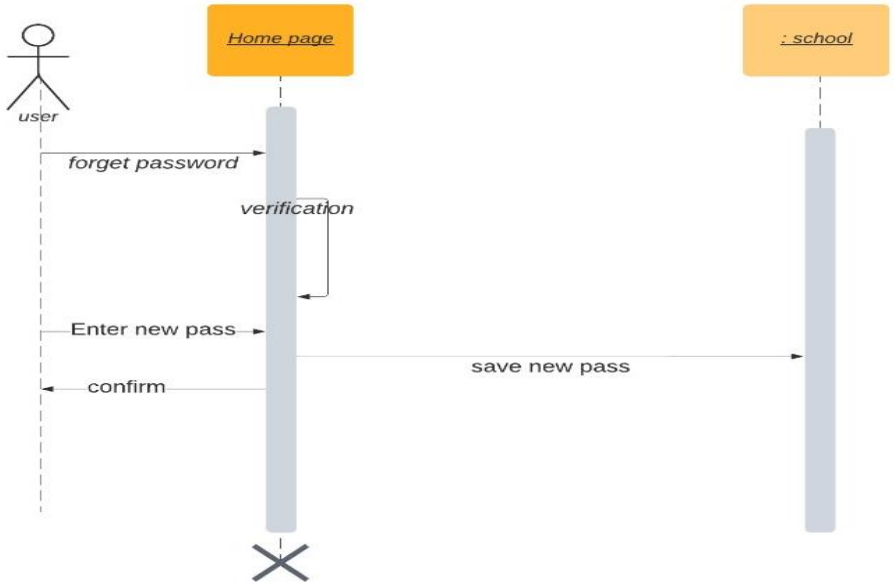
User Login:



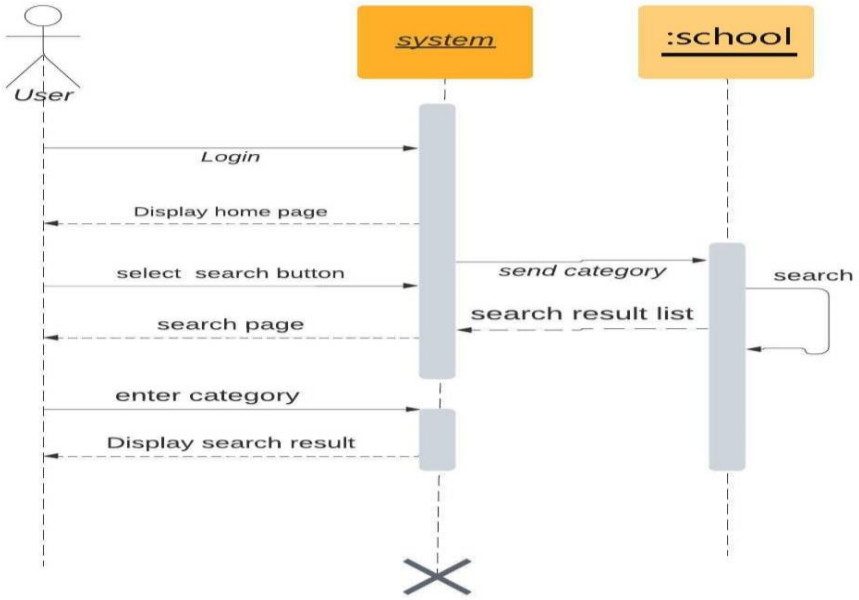
User Forget Password



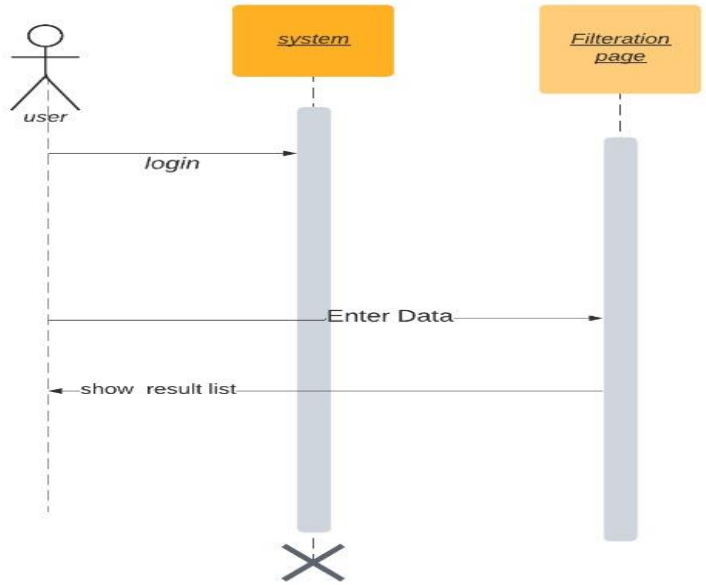
School Forget Password:



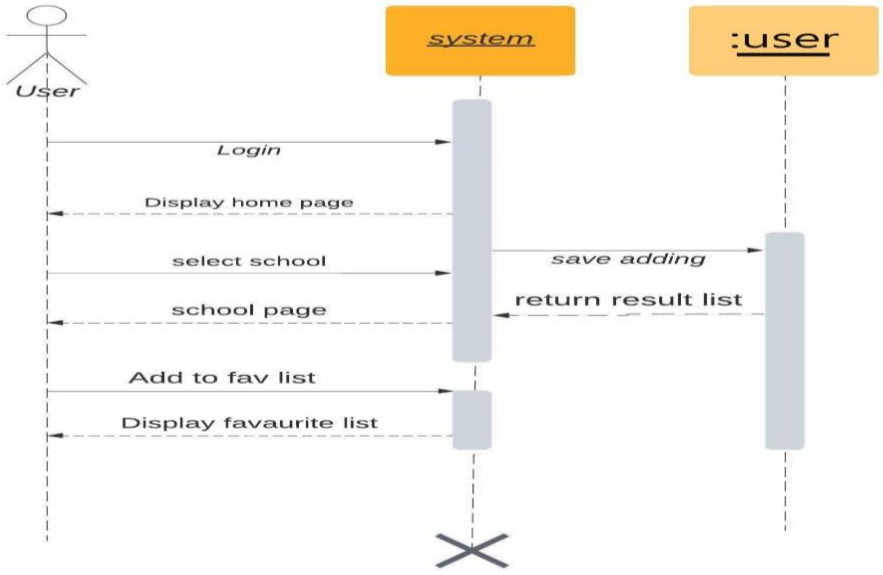
Search



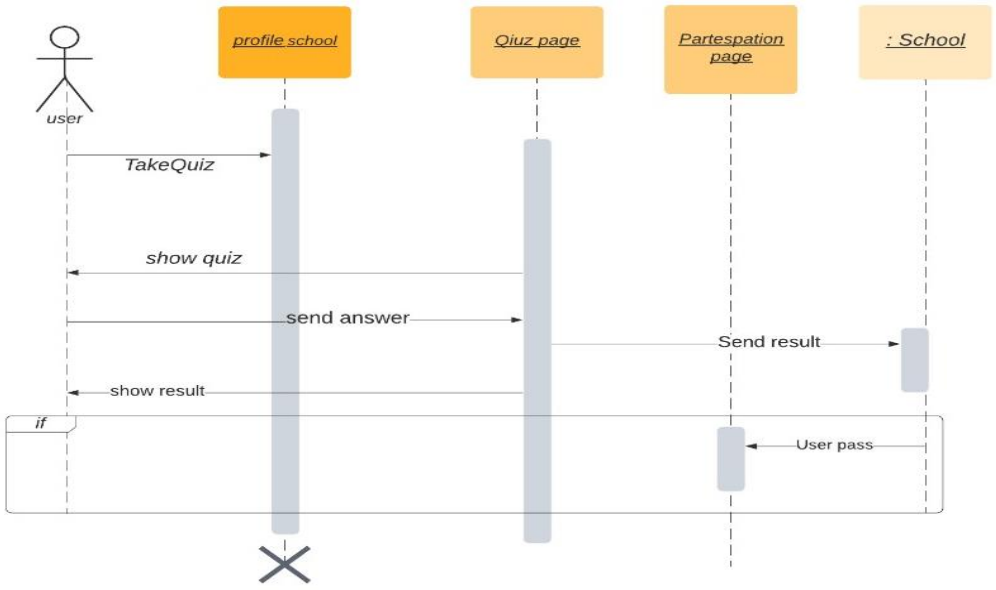
Filtration:



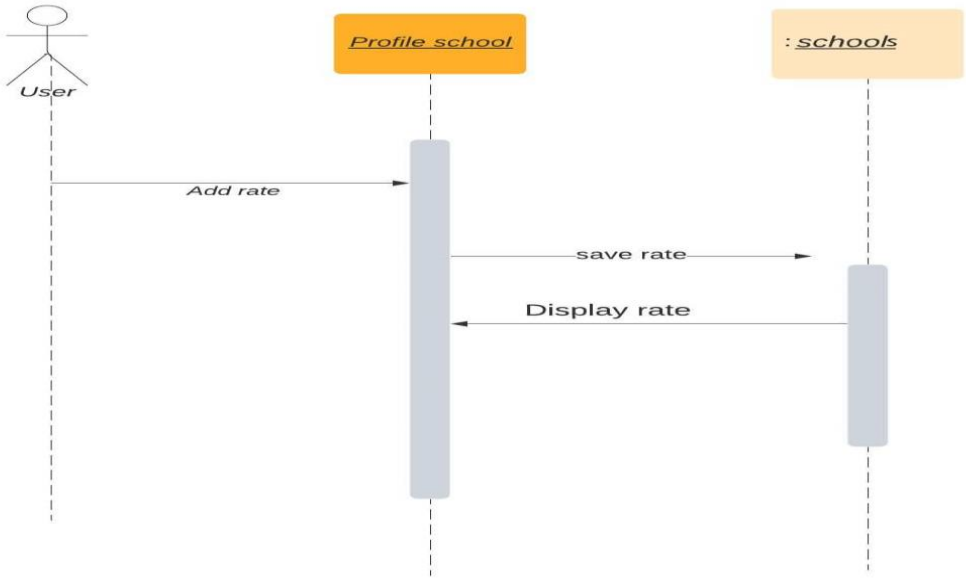
Add to Favorite



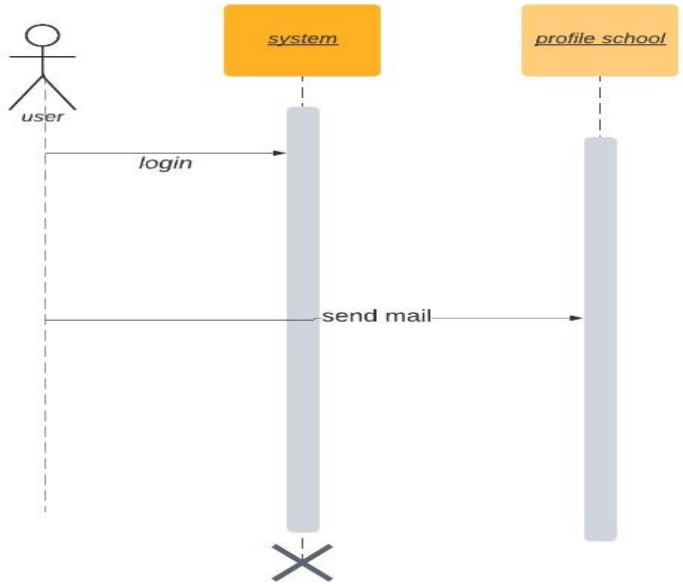
Take Quiz



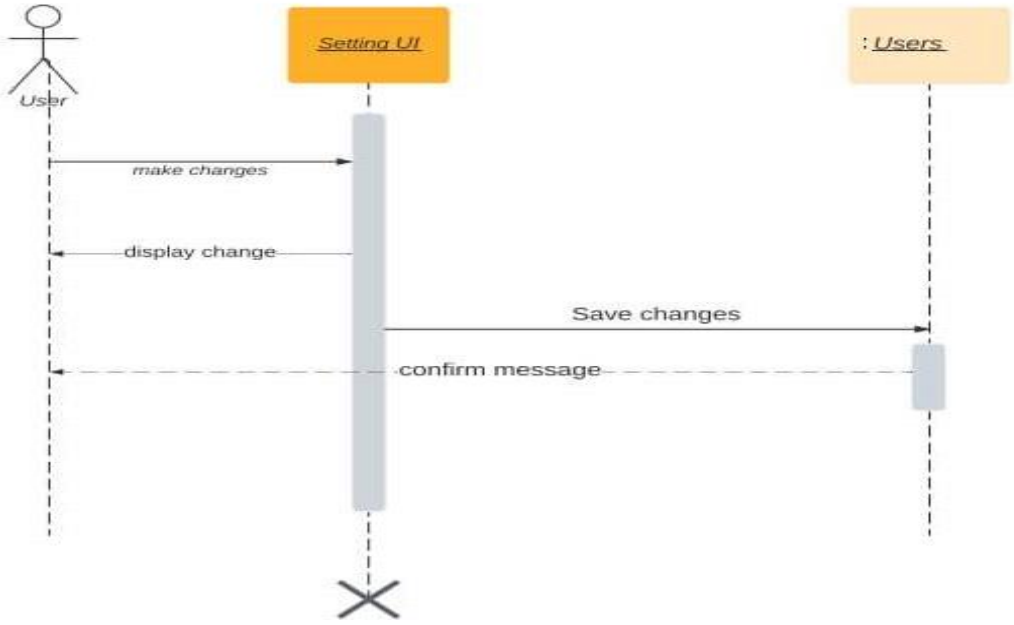
Rate



Send mail



User Edit:



CHAPTER 4

RESULT AND DISCUSSIONS

Results and Discussions chapter should provide the reader with the discussions on the collected results and/or achieved final product capabilities as the result of the graduation project. In this chapter we're going to find out the results of the project whether they're achieved or not and the differences between the desired results and the actual ones.

4.1 Results:

4.1.1 Expected Results:

- ✓ Both Users and Schools can log into the application, Register, and logout.
- ✓ Both Users and schools can Edit and show their profiles.
- ✓ Both Users and Schools can use forget password feature.
- ✓ Users can search for a school.
- ✓ Users can enter their data in Filtration Page to make it easy for the application to get the most similar school for the data which the user entered.
- ✓ Users can add a school to a favorites list then this user will be as a follower of the school. He will be added to the followers' list.
- ✓ Users can send mails to a school.
- ✓ Users can rate any school he wants.
- ✓ Users can take a look at a school's profile and vice versa.
- ✓ Users can take a quiz, answer it, and send the answers to the school.
- ✓ Schools can show their participants.
- ✓ Schools can create quiz.
- ✓ Admin can Accept or Reject school.
- ✓ Admin can delete school.
- ✓ If the user enters some data about a lot of specifications in his mind, the application will show the schools which have accounts and have the same specifications that the user enters.
- ✓ The application has light and dark mode.
- ✓ Both Users and Schools have notifications.
- ✓ Both Users and schools can chat each other.
- ✓ There is a private sector for the schools that support students with special needs.

4.1.2 Actual Results:

- ✓ Both Users and Schools can log into the application, Register, and logout.
- ✓ Both Users and schools can Edit and show their profiles.
- ✓ Both Users and Schools can use forget password feature.
- ✓ Users can search for a school.
- ✓ Users can enter their data in Filtration Page to make it easy for the application to get the most similar school for the data which the user entered.

- ✓ Users can add a school to a favorites list then this user will be as a follower of the school. He will be added to the followers' list.
- ✓ Users can send mails to a school.
- ✓ Users can rate any school he wants.
- ✓ Users can take a look at a school's profile and vice versa.
- ✓ Users can take a quiz, answer it, and send the answers to the school.
- ✓ Schools can show their participants.
- ✓ Schools can create quiz.
- ✓ Admin can Accept or Reject school.
- ✓ Admin can delete school.
- ✓ If the user enters some data about a lot of specifications in his mind, the application will show the schools which have accounts and have the same specifications that the user enters.

4.2 Discussions:

- **Functional Testing:**
  - ✓ Unit Testing:
    A type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

  - ✓ Integration Testing:
    A level of software testing where individual units is combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

  - ✓ System Testing:
    Testing a system behavior when development is done, and the system can be tested as complete entity.

- **Test Cases:**

  - ✓ Registration test case:

Step	Action	Expected Result	Android
Prereq.	Application is already opened		
1	**Register using valid Student credentials**		
1.1	Click on Student button	The Student button is selected	Pass
6.2	click on New account register button	- go direct to student's registeration	Pass
1.2	Enter a valid data		Pass
1.5	Click on Register	-Sign up is successful -User is redirected to home page	Pass
2	**Register using valid School credentials**		
2.1	Click on School button	The School button is selected	Pass
6.2	click on New account register button	- go direct to school 'sregisteration	Pass
2.2	Enter a valid data		Pass
2.5	Click on sign up	-School is waiting to admin accept request registeration - Go redirected to login page	Pass
6	**Sign Up without choosing type of user**		
6.1	leave the user type field without click		Pass
6.2	click on New account register button	- go direct to student's registeration because by default select student button	Pass
11	**Sign Up with in different language**		
10.1	Enter an arabic data		Pass
10.2	fill all other fields with valid data		Pass
10.3	click on sign up	-Sign up is successful -User as student is redirected to home page -User as school is redirected to admin in waiting page	Pass
13	**Sign Up with different password in password field & confirm password**		
13.1	Enter different password in password field & confirm password		Pass
13.2	fill all other fields with valid data		Pass
13.3	click on sign up	An error message appears telling the user that "password does not match"	Pass
15	**Sign up with password less than 6 numbers or characters**		
15.1	Enter pasword less than 6 numbers or characters		Pass
15.2	fill all other fields with valid data		Pass
15.3	click on sign up	An error message appears telling the user that "the password must not be less than 6 numbers or letters"	Pass
17	**Sign Up with empty any field**		
17.1	leave any field empty		Pass
17.2	fill all other fields with valid data		Pass
17.3	click on sign up	An error message appears telling the user "please complete the data"	Pass

✓ Login test case:

	Action	Expected Result	Android
Prereq.	-User Already downloaded the application -User already in login page to the application		
1	**Login with a valid email address**		
1.1	Enter a valid Email Address		Pass
1.2	Enter a valid password		Pass
1.3	Click on "Login"	-User is logged in successfully	Pass
2	**Login with an unregistered email**		
2.1	Enter an unregistered Email		Pass
2.2	Enter a password		Pass
2.3	Click on "Login"	- An Error message that "email not found " appears	Pass
3	**Login with a valid email & an invalid password**		
3.1	Enter a valid Email		Pass
3.2	Enter an invalid password		Pass
3.3	Click on "Login"	- An Error message that "Password is wrong" appears	Pass
4	**Verify that the SHOW/HIDE button works**		
4.1	Enter an Email address		Pass
4.2	Enter a Password		Pass
4.3	Click on "SHOW" icon	- The password is shown to the user	Pass
4.4	Click on "HIDE" icon	- The password is encrypted to the user	Pass
5	**Verify rotating the screen during logging in**		
5.1	Enter a valid Email Address or username		Pass
5.2	Enter a valid password		Pass
5.3	Rotate the device screen from landscape to portrait mode	-Application rotates normally and fields are still filled with previously entered data	Pass
5.4	Click on "Login"	-User is logged in successfully	Pass
6	**Verify forget password functionality using email**		
6.2	Enter a valid email		Pass
6.1	Click on Forget Password	-A message appears telling the user that " check your email please"	Pass
6.3	check my gmail	-A message appears telling the user that a message is sent to his email	Pass
6.4	Open the message that is sent to the email		Pass
6.5	change old password		Pass
6.6	Enter new password	User can login normally using the new password	Pass

- ✓ School's home test case:

Step	Action	Expected Result	Android
Prereq.	-User already in Home page		
1	check on Profile button		
1.1	click on profile button	appear school profile	Pass
2	check on Quiz button		
2.1	click on Quiz button	appear create quiz page	Pass
3	check on ShowQuiz button		
3.1	click on ShowQuiz button	appear quiz that is created	Pass
4	check on participants button		
4.1	click on participants button	appear students that passed in quiz	Pass
5	check on followers button		
5.1	click on followers button	appear students that is loved on my school	Pass

- ✓ Filtration test case:

Step	Action	Expected Result	Android
Prereq.	-User already in Filteration page		
1	Check if the filtering leads to right results		
1.1	Enter all required Data		Pass
1.2	click on Filter button	appear right results that match data 's entered	Pass
1	Check if the clicking on the filter button opens filter page		
1.1	open Drawer		Pass
1.2	click on Filteration	appear Filteration page to enter data that is required	Pass

- ✓ User's home test case:

Step	Action	Expected Result	Android
Prereq.	-User already in Home page		
1	check on drawer		
1.1	click on drawer	drawer open	Pass
2	check on search icon		
2.1	click on search icon	appear field that allow search on any school i wanted	Pass
3	click on any school in home page		
3.1	click on any school	appear profile school &quiz&send mail &rating	Pass
4	scroll down in home page		
4.1	scroll down in home page	appear all exist schools on application	Pass

- ✓ Search test case:

Step	Action	Expected Result	Android
Prereq.	-User already in search page		
1	Search on exist school		
1.1	Enter characters that is schoolname	appear school direct	Pass
2	Search on not exist school		
2.1	Enter characters that is schoolname	doesn't appear any thing	Pass

- ✓ Rate test case:

Step	Action	Expected Result	Android
Prereq.	-User already in school Profile page as user		
1	check on rating		
1.1	Enter my rate on school	save user rate and hide container rating	Pass
2	change rate		
2.1	Enter my new rate on school	save user rate and hide container rating	Pass

- ✓ Quiz test case:

Step	Action	Expected Result	Android
Prereq.	-User already in create Quiz page		
1	check on true & false Quiz		
1.1	select T&F	appear true & false page	Pass
1.2	Enter Question		Pass
1.3	Select answer		Pass
1.4	click on add button	go to create next question	Pass
1.5	click on back arrow	go to home page	Pass
2	check on choice Quiz		
2.1	select choice	appear choice page	Pass
2.2	Enter Question		Pass
2.3	Enter answers		Pass
2.4	Select right answer		Pass
2.5	click on add button	go to create next question	Pass
2.6	click on back arrow	go to home page	Pass
Prereq.	-User already in show Quiz page		
1	check on show Quiz page in exist quiz		
1.1	click on show Quiz button	appear showQuiz page that is created	Pass
2	check on show Quiz page in not exist quiz		
2.1	click on show Quiz button	appear showQuiz page empty	Pass
Prereq.	-User already in take Quiz page		
1	check on take Quiz button		
1.1	click on take Quiz button	appear Quiz page that is created	Pass
1.2	Solve Quiz		
1.3	click on send button	appear result & appear correct answers	

- ✓ Participants test case:

Step	Action	Expected Result	Android
Prereq.	-User already open Drawer		
1	check on particpant button		
1.1	click on participant button	go to direct participant page & appear users that passed in quiz	Pass

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

The Conclusion and Recommendation chapter conclude the report by stating the task, the difficulties faced, experiences gained, results achieved and final thoughts on the project. Then Recommendations for future work and project enhancement should be included.

5.1 Conclusion:

Finally, we did our best to get a perfect result. We also want to serve a sector of our society. We choose the students and their parents. We want to save money, time, and efforts for choosing a good school.

A lot of students miss the year because they didn't find a suitable school for their budget and their conditions.

A lot of schools spend their money on marketing to find good students that accept their instructions.

We hope we can help all of those people to find each other quickly without wasting time or money.

We made a bridge between the user and the school to communicate with each other and to get much knowledge.

Although we faced some difficulties such as:

- Studying a new technologies and techniques.
- Fixing errors and learning how to avoid repeating that mistake.
- Learning time management.
- Set deadlines and Committed them.
- Get free resources.

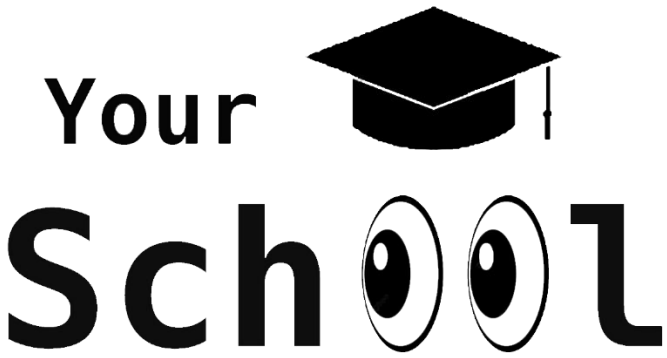
We gained much experience like:

- Time management.
- Teamworking.
- Searching about a certain topic or a certain tutorial.
- Putting a plan.

- Dividing the tasks into sprints.
- How to apply a new technology.

5.2 Recommendations for Future Work:

- We will make a private sector to the student with special needs' schools.
- Add a notification feature.
- Add a chat between the user and the school.
- Active the dark mode.



Thank You.