Analysis and design of talent development prototype (Gifted)

Graduation Project, Part-I (SWE 496)
Software Engineering Department
CCIS, KSU

Project Advisor:
Dr.Ahmed Ghoneim

by
Abdulaziz Alqudayri, 441102562
Faisal Aljohani, 441101642
Nawaf Alothman, 441110448
Abdulrahman Abuabat, 441102274

List Of Figures

List Of Tables

Abstract

Talent development is a complex task, we aim to build a system that talents can use to figure out their talent, develop their talent, and find opportunities based on their talent. The system will make use of acceptance tests, check the level of the talent, then help the talent improve their skill, and connect the talent to a third-party organization that he can cater his/her Skills. For these third-party organizations, finding the right talented people can be a hard task. Our aim is to bridge the process between organizations and talents by providing a talent pole which can be viewed and picked from. A sad sight in today's community, is the waste of talents. We believe not enough talents are being capitalized on.

1 Introduction

1.1 Introduction

We aim to build a software to help talented individuals discover and grow their talents. The software will also help talents benefit from their talents. Gifted, the name refers to every talented individual that either wants to develop their talent, or doesn't even know of their talent yet. We chose the name in the memory of all the wasted talent that couldn't find a place to incubate their talent.

1.2 Background

Talent is not restricted to one kind or shape. We believe talent comes in all shapes, from poetry to mathematics. The problem with talented people is mostly direction, a child with a gift in mathematics' journey is always made easier if he indulges in a community of people with similar interests. To achieve such goal, one must start with making sure everyone in said community is interested in the community theme. People come from different backgrounds, and talented individuals who were raised in a community that revolves around that talent will always have an advantage, and they will always do. Our goal is to reduce this imbalance of opportunity. We want to give every talent the chance to shine, and pursue their interests.

2 Domain analysis

In the following section, 8 different organizations will be analyzed, each organization is similar to the software we're building in a certain way. Analyzed organizations are all different and have different goals from one another. This analysis covers organizations from all over the world, including organizations in the MENA region.

2.1 Scalers

Scalers are a B2B sales talent development, they only offer help improving sales skills. Their process consists of a 3 week course, consisting of exams, mock sales, and lectures. Scalers are on a mission to change the sales landscape of B2B sales in Saudi. Big part of what makes Scalers special is, they link their students with companies and employers. Such that, whenever someone graduates from their course, they help him find a job[1].

2.1.1 Features

- View grades
    - Students are graded throughout the course, and they can follow their progress through the grades tab as shown in Fig.2.2 .
- LMS
    - Scalers have their own learning management system, where all their students get access to their exams, and lectures.
- Authentication
    - Talents must provide credentials to login to the platform.
- Live lectures
    - Scalers offer live lectures to students, they do it through integrating their system with Zoom software.
- Referral program
    - Each student is given a code they can share with their network, which earns them credit to use on the platform.
- Integrated with BNPL
    - The talent can use BNPL (Buy now pay later) technology to cover the costs of the course.

2.1.2 Constraints

- Cohort based
    - Scalers are cohort based, and students can only apply during a cohort opening
- No acceptance tests
    - Scalers don’t conduct acceptance tests on applicants to measure the talents
- Small segment
    - Scalers only cover sales talents, no other type of talent is supported
- Slow
    - The scalers system is built on no-code technology, which leads to slow and poor performance as shown Fig.2.1 .

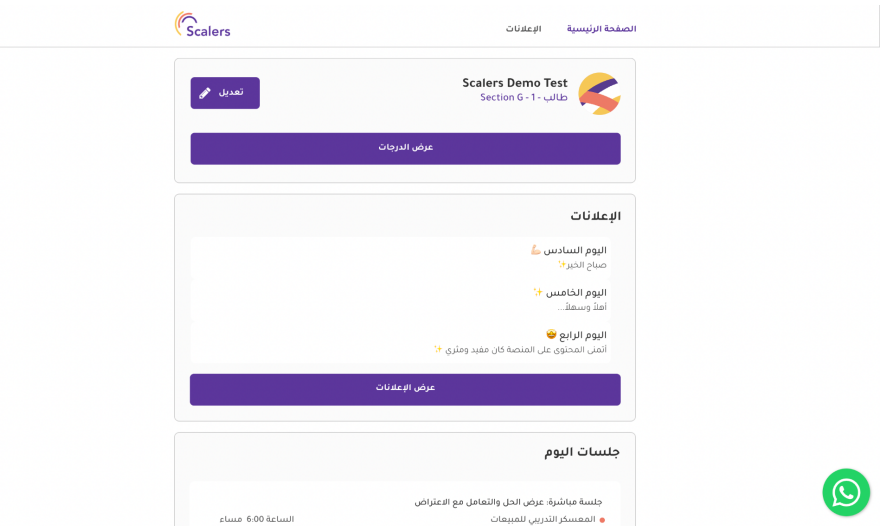


Fig.2.1 Scalers student dashboard

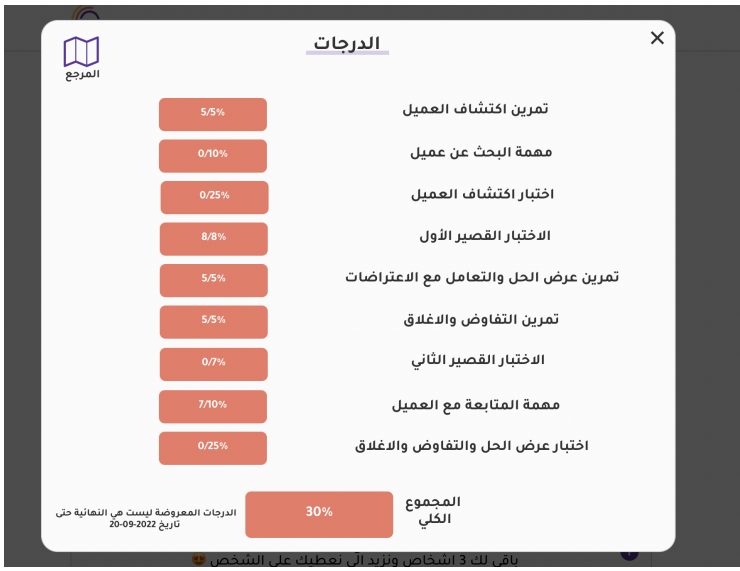


Fig.2.2 View grades

2.2 Talented Athlete Scholarship Scheme

TASS is a sport partnership between talented athletes, education institutions and the national governing bodies of sport as shown in Fig.2.3 TASS aims to identify, develop and fund those athletes, it helps with providing lifestyle and well-being support for athletes in education. TASS supports more than 500 Athletes in 38 sports in England[2].

2.2.1 Features

- Training
  - TASS provides training for their athletes, so that they can elevate their level of athleticism
- Portal
  - TASS has a portal for their athletes to view their progress and document their progress
- Diverse sports

    - TASS covers a wide variety of sports, such as table tennis
- Handicapped people support
    - TASS also assists with handicapped people, their athletes participate in the paralympic games.
- Workshops
    - TASS hosts a number of workshops/events throughout the year

2.2.2 Constraints

- Sports only
    - TASS only helps with sport talents
- Age restrictions
    - TASS only helps with students above the age of 16
- Nationality restrictions
    - TASS only helps with British students
- No application process
    - TASS doesn't have an application process. Instead, they only take students nominated by their National Governing Body.

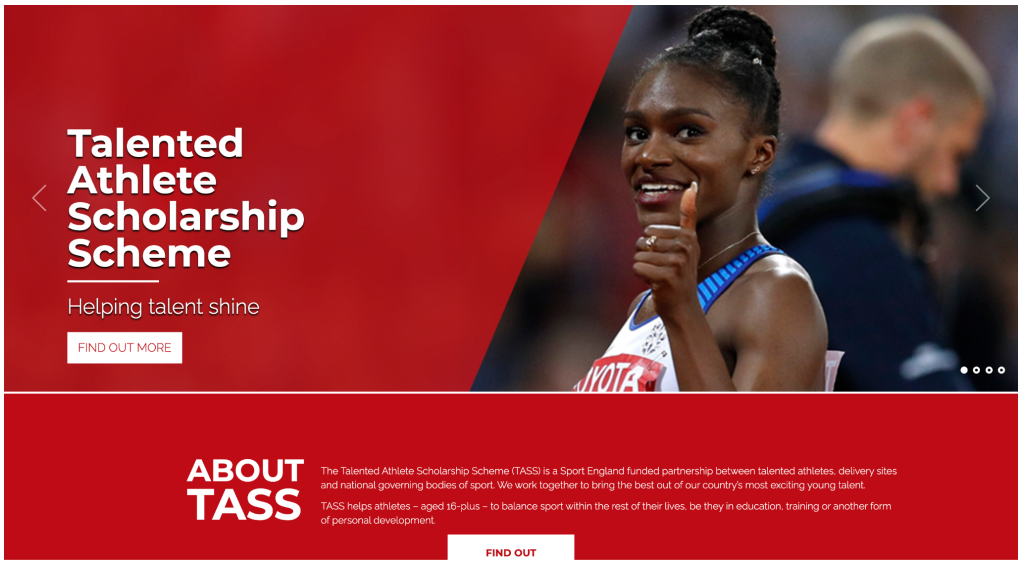


Fig.2.3 Talented Athlete Scholarship homepage

2.3 Qimam fellowship

Qimam is a fellowship program that aims to identify, develop and empower the most promising and distinguished university students in and from Saudi Arabia to achieve their full potential. Their vision is to be part of the development of these high potential candidates and provide them with the necessary tools to success[3].

2.3.1 Features

- Leadership training
    - The program is supported by a coalition of leading national and international companies.
- Mentorships
    - They provide one-on-one mentorship from senior public and private sectors leaders.
- Career opportunities
    - Shares profiles of fellows with leading HR executives in Saudi Arabia to facilitate their career opportunities.
- Well designed website
    - The website looks simple and easy to use as shown in Fig.2.4 .
- Software application form
    - Students can provide their information using a form on the website as shown in Fig.2.5 .
- Free of charge.
    - All the benefits are completely free of charge.
- Alumni network
    - They provide the names and contacts of all their graduates to the public, they also create a private community to their alumni

2.3.2 Constraints

- Strict evaluation process
    - To be admitted, candidates have to go through a very rigorous evaluation process.

- Program density
    - The program is very dense over 12 days only.
- Limited segment.
    - Only university students are accepted.

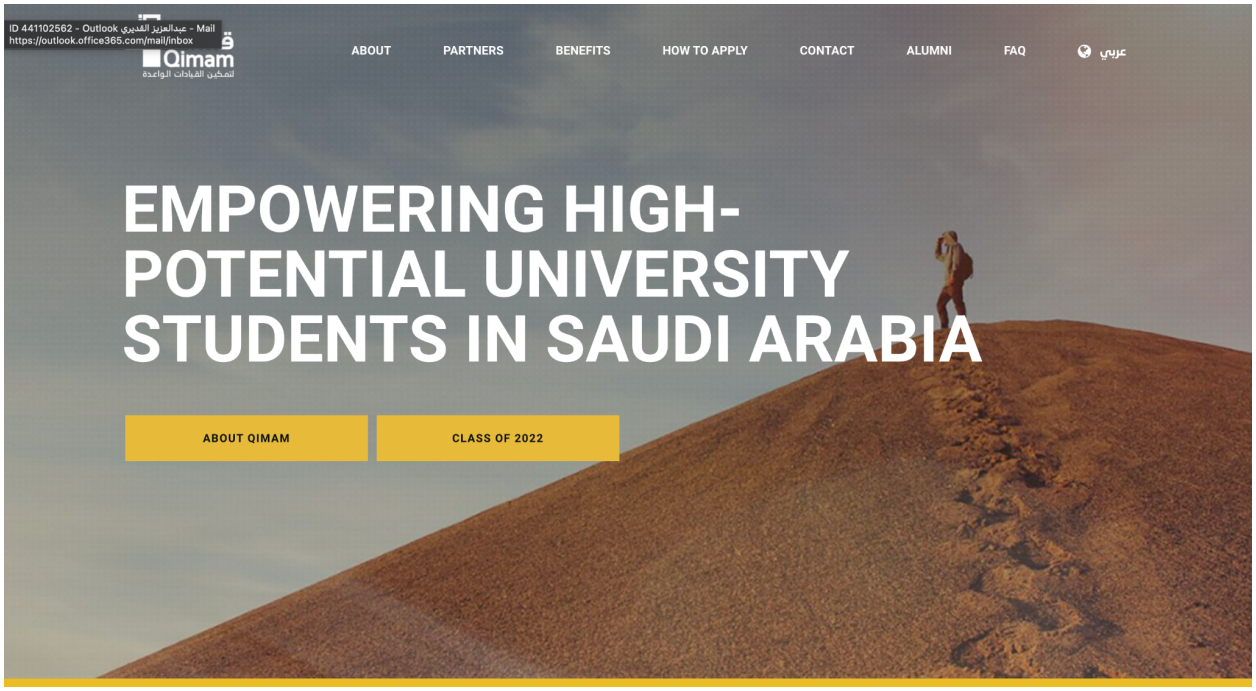


Fig.2.4 Qimam homepage

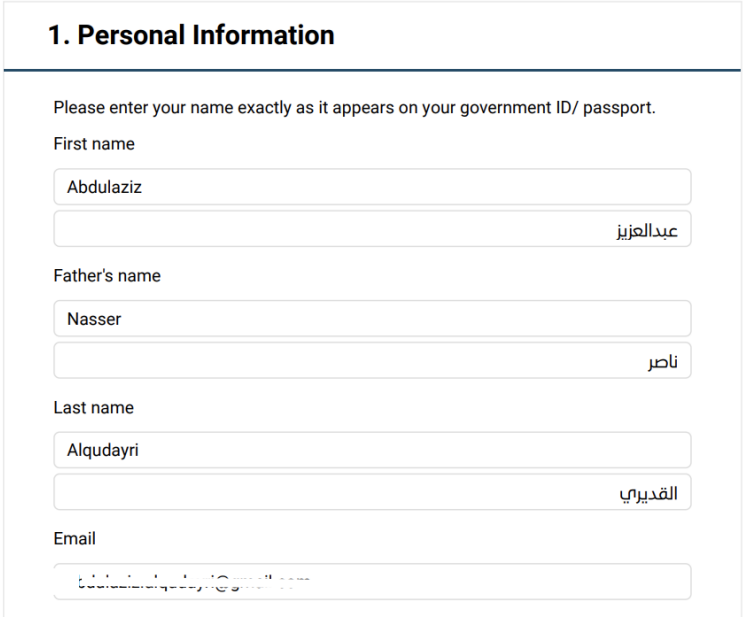


Fig.2.5 Qimam application form

2.4 Institute for Educational Advancement

IEA is a nonprofit organization. They rely on the support of foundations, corporations and individuals to provide programs and services for young gifted students. Their goal is to ensure that each gifted child's specific needs are met so that they can work toward meeting their full intellectual and personal potential[4].

2.4.1 Features

- Online donations
  - You can donate directly from their website as shown in Fig.2.6 .
- Newsletter subscription
  - By signing up, you will receive emails about their latest news.
- Blogs

    - They have blogs regarding education and gifted students.

2.4.2 Constraints

- Pages missing
    - several missing pages as shown in Fig.2.7 .
- Heavy website
    - The website is loaded with too many pages.
- Limited segment
    - They only serve students under 18 years old.

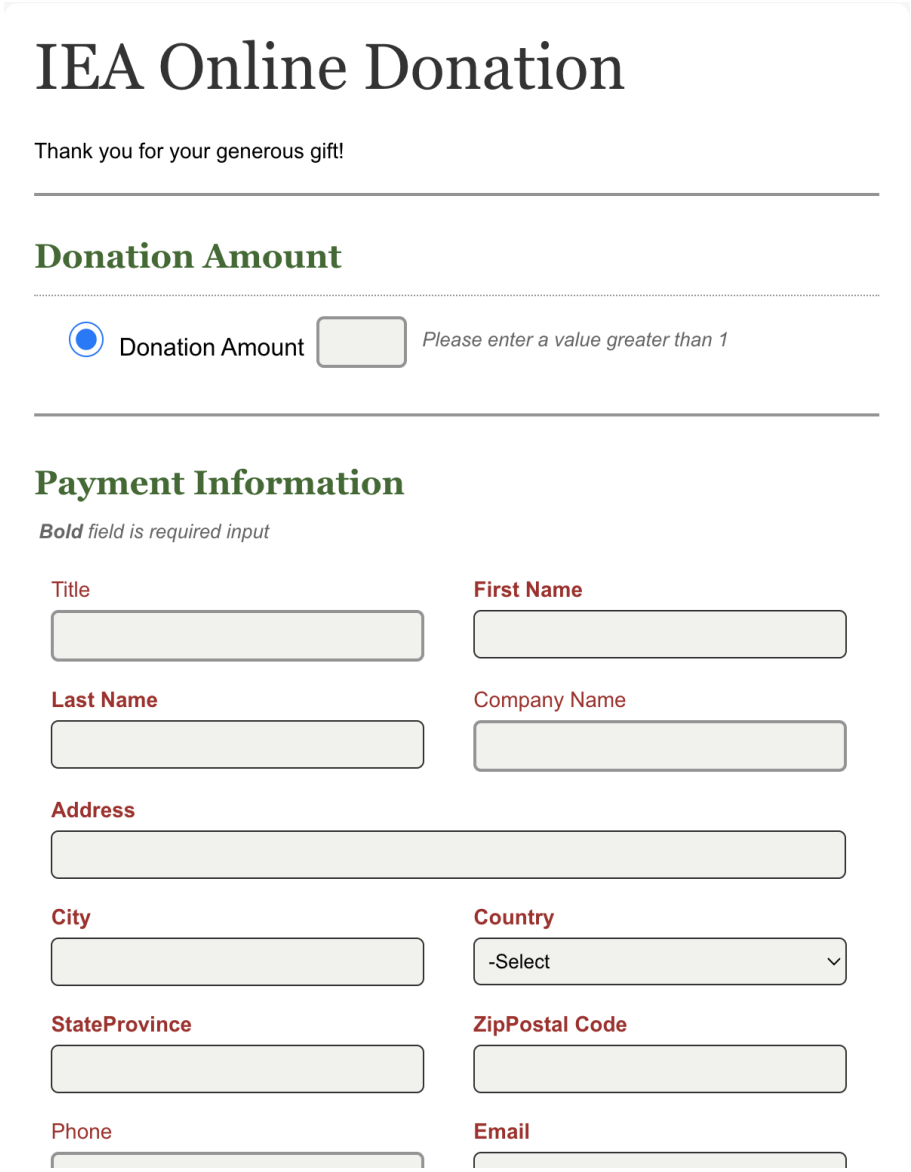


Fig.2.6 IEA Online Donation

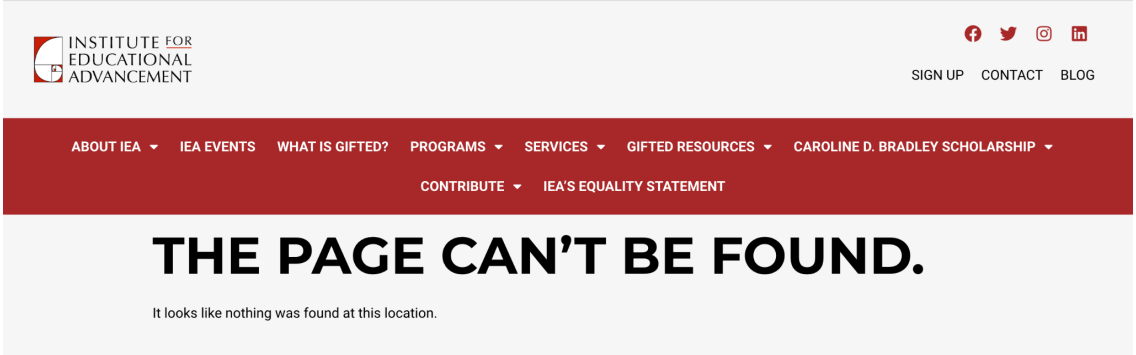


Fig.2.7 IEA page not available

2.5 Mawhiba

King Abdulaziz and his Companions Foundation for Giftedness and Creativity "Mawhiba" is a non-profit endowment organization that aims to help talented kids to grow and to provide them the knowledge that they need, Mawhiba's targeted segments are school students to help them to grow by providing them the best environment to grow as shown in Fig.2.8 [5].

2.5.1 Features

- Benefits For Students
    - Mawhiba students get special treatment from schools all over Saudi.
    - Mawhiba connects its students with researchers to work on professional research.
    - Mawhiba provides for the students the ability to participate in different international science competitions as shown in Fig.2.9 .
- Non-Profit
    - It's a non-profit endowment organization that is supported by the government.
- Partners
    - Mawhiba has a large number of partners in different fields(Ministry of Education, Aramco, King Saud University).
- Awards.
    - Award of the Arab Bureau of Education for the Gulf states.
    - Khalifa Education Award.
- App Notifications
    - Mawhiba keeps you Up-To-Date with new courses and new opportunities through App Notifications.
- Live Support
    - Mawhiba Provides live chat support to help the user.

2.5.2 Constraints

- Limited segment.
    - Very wide range of Mawhiba's programs are for school students.
    - Most of the students are nominated, not the other way around.
- App Performance
    - Mawhiba's App suffers from bugs and lack of exception handling.
- Tests
    - Mawhiba doesn't provide online tests for students, they have to visit Mawhiba's Headquarter to conduct them

Fig.2.8 Student record

Fig.2.9 Mawhiba homepage

2.6 Johns Hopkins Center for Talented Youth (CTY)

The Johns Hopkins Center for Talented Youth is a non-profit organization that aims to provide the best suited environment for pre-college students by giving the students tests and courses to choose the major that they are interested in, after choosing the major, CTY will help the students to develop the skills that is required for being the best in this major[6].

2.6.1 Features

- Programs
    - CTY provides a high number of different courses in different fields for gifted youth.
    - CTY Helps students with choosing an academic career with their Academic Exploration.
    - CTY provides online courses and summer courses that will take place on Campus.
- Supervision
    - CTY provides for student's parents the ability to track their children's progress in courses by providing them with an account on their website.
- Tests
    - Students can take tests online on CTY's website or submit their test scores as shown in Fig.2.10 .

2.6.2 Constraints

- Mobile Application
    - CTY doesn't provide mobile application as shown in Fig.2.11 .
- Organizations
    - CTY doesn't link students with different organizations.
- Paid membership
    - Since there is no organization to fund CTY, parents of the students need to pay fees for registering their childerens.
- Limited Segment:
    - CTY only accepts students from grade 2 - 12.

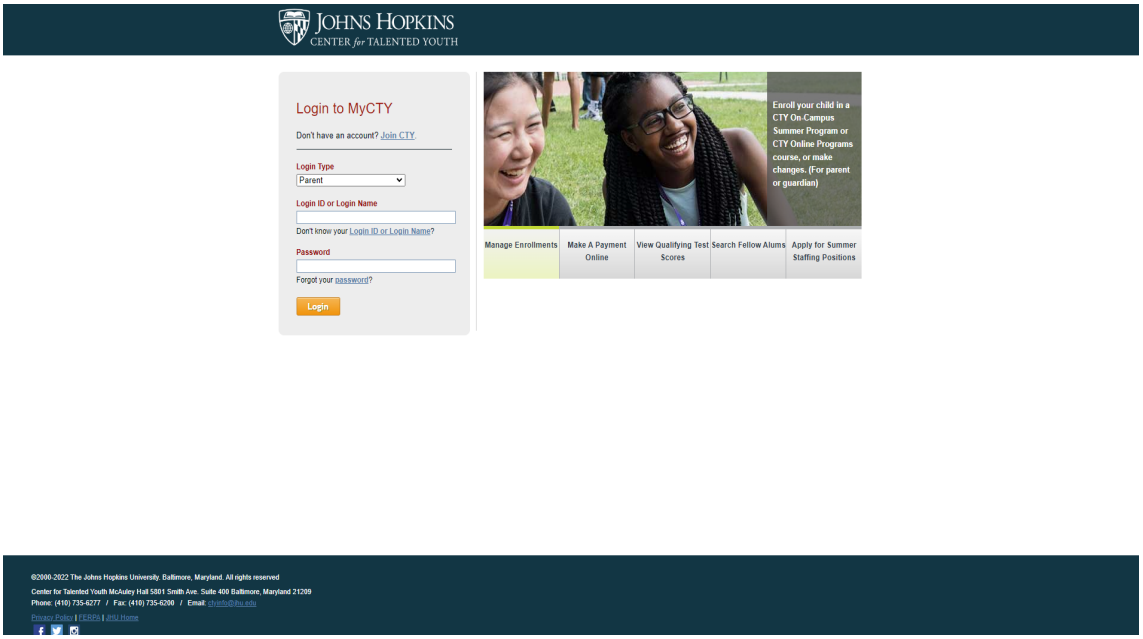


Fig.2.10 JOHNS HOPKINS login page

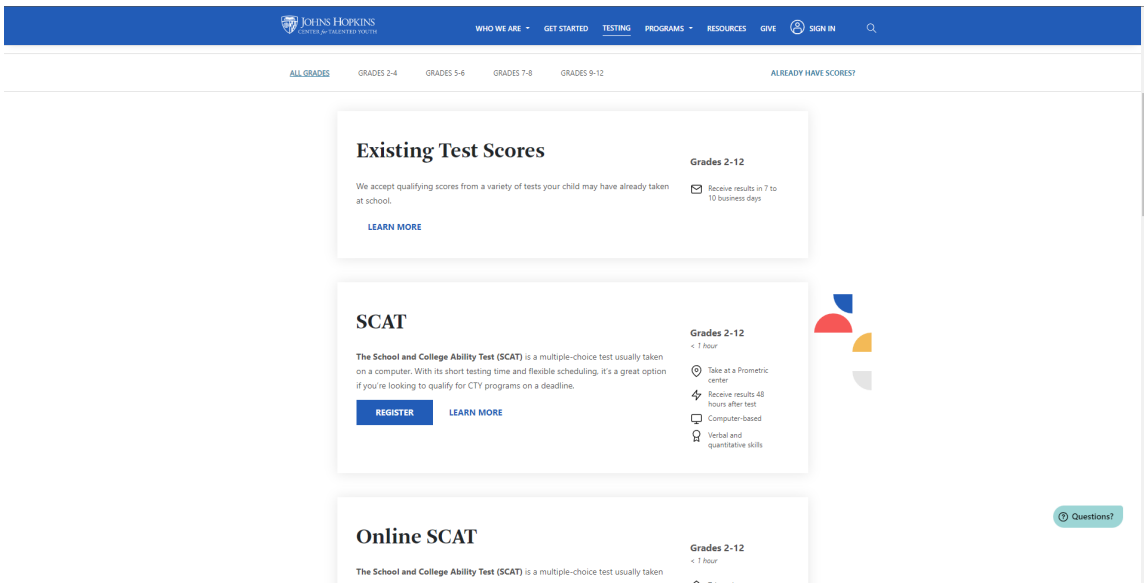


Fig.2.11 JOHNS HOPKINS Grades page

2.7 1 Stalent

1stalent specialize in skills acquisition. They provide a wide range of solutions including direct hire, outsourcing, and other services. The industries they work in include Retail, Health care, Consulting, and construction. 1stalent supports the implementation of professional and ethical talent acquisition[7].

2.7.1 Features

- Sourcing service
    - This service uses the latest technology to reduce the time consumed traditionally by recruiters. This service also removes human bias and focuses on finding the best candidate.

- Chatbot technology
    - They use Artificial intelligence and chatbot technology to ensure that they source the best candidate to suit their clients as shown in Fig.2.12 .
- Software application form
    - The third parties are given a form to fill in what skills they are seeking for, and from the pole, a person who fits their needs will be selected for them as shown in Fig.2.13.

2.7.2 Constraints

- No scouting for talents
    - In order to showcase talented people's ability, they must be the ones to apply.
- They don't cover a wide variety of talent types
    - There are talents that are not focused on such as sport and poetry.

- No training program
    - Applications don't get coached nor get the instructions that can help them get fit for that job.

- No acceptance test
    - They are assessed by the form they fill out and the chatbot.

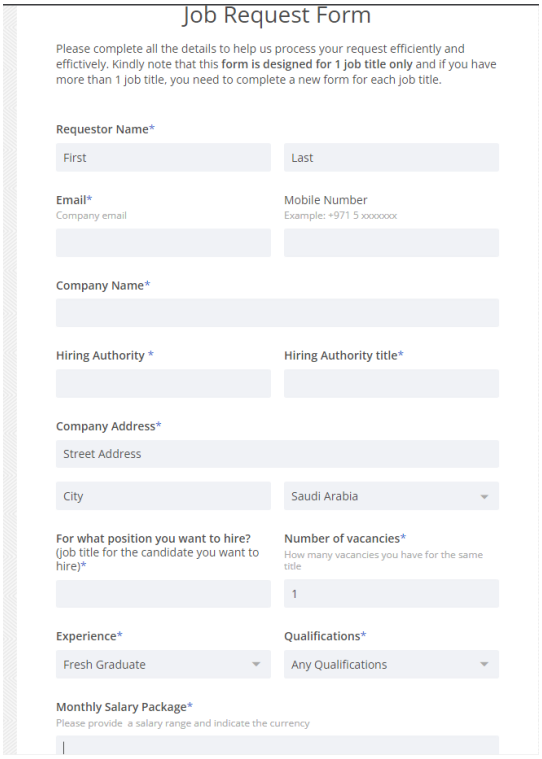


Fig.2.12 Job request form

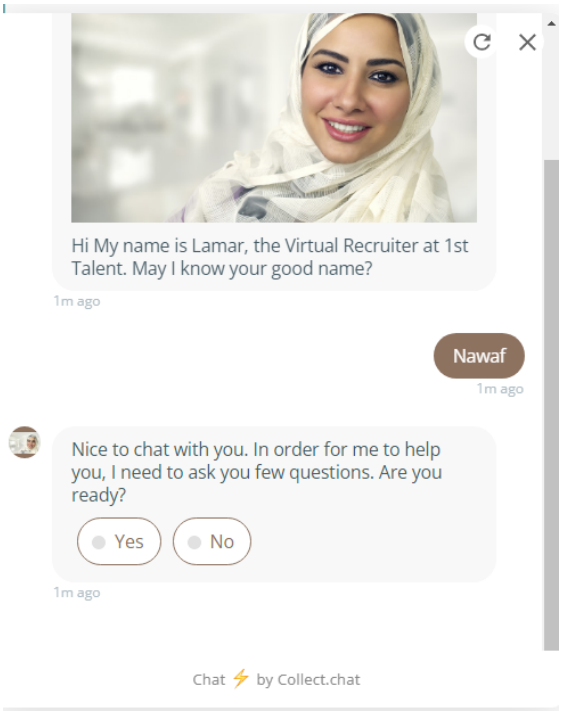


Fig.2.13 AI Chatbot

2.8 MARKODING

Markoding is a non-profit organization based in Indonesia. Their aim is to help unprivileged youth in Indonesia to become a generation of inventors by providing them with 21st-century skills. Stopping youth unemployment by bridging the skills gap[8].

2.8.1 Features

- On-demand digital skills
    - Through an on-demand program, adolescents will learn about various digital skills such as game development, web and game design, and product management. They will also develop a digital product that they will use as their portfolio. To get the most out of this program, students need to create a total of over a hundred words. For the best predictions, they should paste in at least five paragraphs.

- Social Learning Platform
    - The goal of this program is to provide adolescents with a social learning platform that will allow them to develop their digital skills. It features a variety of learning methods, such as live workshops, online classes, and peer learning as shown in Fig.2.14 .

- Digital innovation challenges
    - Through a partnership between UNICEF Indonesia and the game development company, Markoding, the program aims to provide adolescent boys and girls with the necessary skills to succeed in their future careers.

2.8.2 Constraints

- They don’t cover a wide variety of talent types
    - They are only focused on the technical side.
- No acceptance test
    - They are assessed by the form they fill out and the chatbot.

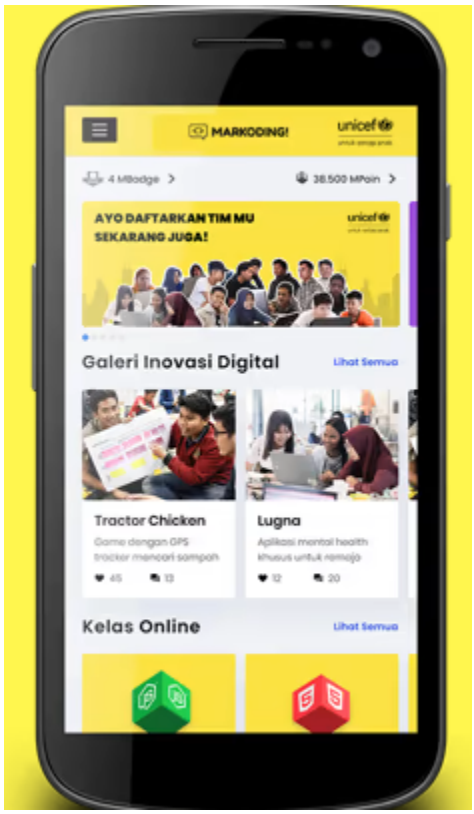


Fig.2.14 Markoding app land page

2.9 Analysis

To build our system, we decided to combine the discovered features from the conducted domain analysis, and add our own twist to it. The table below (Table 2.9.1) summarises the features we discovered from our domain analysis, each column represents a certain entity, and the rows represent the discovered features. The last column represents our system (Gifted), Gifted doesn't have every single feature discovered in our domain analysis, but a combination of what we believe to be the most important features.

	Scalers	TASS	1stalent	MARKODING	Qimam	IEA	Mawhiba	CTY	**Gifted (Our System)**
Tests	✓				✓		✓	✓	✓
Social learning	✓			✓			✓		✓
LMS	✓	✓		✓			✓	✓	✓
Cohorts	✓	✓			✓				
Training	✓	✓		✓	✓	✓	✓	✓	✓
Paid	✓		✓						
Connected with 3rd party	✓	✓	✓		✓	✓	✓	✓	✓
Limited capacity		✓			✓		✓	✓	
Software dependency	✓	✓		✓			✓		✓
Specific specialization	✓	✓			✓				✓
Youth support		✓		✓		✓	✓	✓	✓

Table 2.1 Comparison between Gifted and other available solutions

3 Risks & Constraints

Software projects regularly face risks and constraints all the time, and that never stopped them from achieving their work. We have defined our risks and constraints and we will ensure that they will be detected in the early phases and managed.

3.1 Risks

3.1.1 Inability to find a suitable 3rd party systems

Big part of the software lies within having suitable courses for talented students in various fields.

3.1.2 High complexity

The system contains a high level of complexity, considering different connections with various 3rd party systems.

3.1.3 Uncertainty

The outcome of this project is not guaranteed to achieve business and social goals.

3.2 Constraints

3.2.1 Large Scope

The team has never worked in a project as big as this one.

3.2.2 Limited timeframe

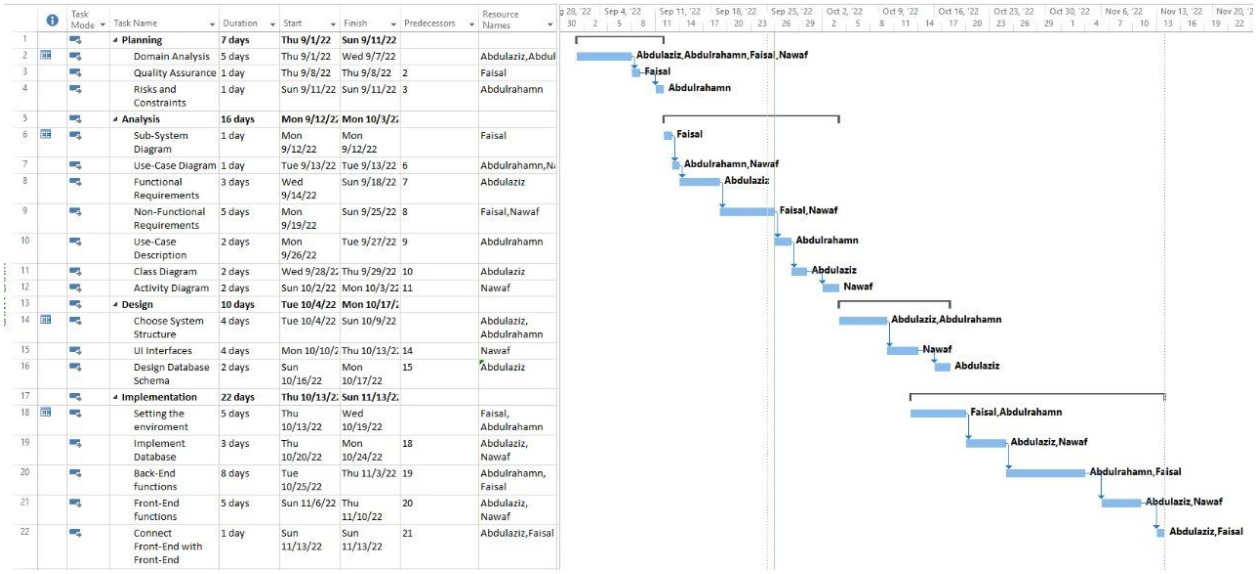
Given the project size and with an inexperienced team, the time is considered limited.

3.2.3 Inexperienced team

The team has never been in a project with this density, and they will have to use new tools and technologies.

4 Project plan

The following Figure 4.1 is the project gantt chart and timeline we will try to follow to achieve our goals, the project is split over 4 phases, Planning, Analysis, Design, and implementation. Each phase is split over multiple tasks and deliverables.



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Planning**	**7 days**	**Thu 9/1/22**	**Sun 9/11/22**		
2		Domain Analysis	5 days	Thu 9/1/22	Wed 9/7/22		Abdulaziz,Abdul
3		Quality Assurance	1 day	Thu 9/8/22	Thu 9/8/22	2	Faisal
4		Risks and Constraints	1 day	Sun 9/11/22	Sun 9/11/22	3	Abdulrahamn
5		**Analysis**	**16 days**	**Mon 9/12/2**	**Mon 10/3/2**		
6		Sub-System Diagram	1 day	Mon 9/12/22	Mon 9/12/22		Faisal
7		Use-Case Diagram	1 day	Tue 9/13/22	Tue 9/13/22	6	Abdulrahamn,N
8		Functional Requirements	3 days	Wed 9/14/22	Sun 9/18/22	7	Abdulaziz
9		Non-Functional Requirements	5 days	Mon 9/19/22	Sun 9/25/22	8	Faisal,Nawaf
10		Use-Case Description	2 days	Mon 9/26/22	Tue 9/27/22	9	Abdulrahamn
11		Class Diagram	2 days	Wed 9/28/2	Thu 9/29/22	10	Abdulaziz
12		Activity Diagram	2 days	Sun 10/2/22	Mon 10/3/2	11	Nawaf
13		**Design**	**10 days**	**Tue 10/4/22**	**Mon 10/17/**		
14		Choose System Structure	4 days	Tue 10/4/22	Sun 10/9/22		Abdulaziz, Abdulrahamn
15		UI Interfaces	4 days	Mon 10/10/2	Thu 10/13/2	14	Nawaf
16		Design Database Schema	2 days	Sun 10/16/22	Mon 10/17/22	15	Abdulaziz
17		**Implementation**	**22 days**	**Thu 10/13/2**	**Sun 11/13/2**		
18		Setting the enviroment	5 days	Thu 10/13/22	Wed 10/19/22		Faisal, Abdulrahamn
19		Implement Database	3 days	Thu 10/20/22	Mon 10/24/22	18	Abdulaziz, Nawaf
20		Back-End functions	8 days	Tue 10/25/22	Thu 11/3/22	19	Abdulrahamn, Faisal
21		Front-End functions	5 days	Sun 11/6/22	Thu 11/10/22	20	Abdulaziz, Nawaf
22		Connect Front-End with Front-End	1 day	Sun 11/13/22	Sun 11/13/22	21	Abdulaziz,Faisal

Fig.4.1 Project plan gantt chart

5 Quality Assurance Plan

The purpose of the Software Quality Plan is to define the techniques, procedures, and methodologies that will be used to assure timely delivery of the software and that the development system meets the specified requirements within project resources. The reason behind creating a software quality plan is because we are concerned with ensuring that the required level of quality is achieved in a software.

5.1 Inspections

After dividing tasks among the team members and once they finish their assigned tasks, they check each other's work and give feedback to the person initially responsible for that task.

5.2 Formal Reviews

Before our meeting with the advisor, every week, we will review the work that we completed. We will also perform a full review of the code that has been pushed to the code base to ensure that it is aligned with the project's requirements and quality standards.

5.3 Verification

We will be creating a checklist for each component of the system, to assure the component developed meets the required criteria. The member responsible for creating and following up with the checklist should never be the same person developing the component.

6 System overview

This section will discuss the system's from an internal point of view. Making use of UML to draw the system, reviewing this section should allow the reader to understand how the system is built on a high level.

6.1 Abstract view

The abstract view diagram seen in figure 6.1 helps understand what are the system's inputs and outputs. From a quick glance at the diagram, one can notice that we will use the talent's age and education level and test scores to assess the talent, we will also decide on the talent maturity.

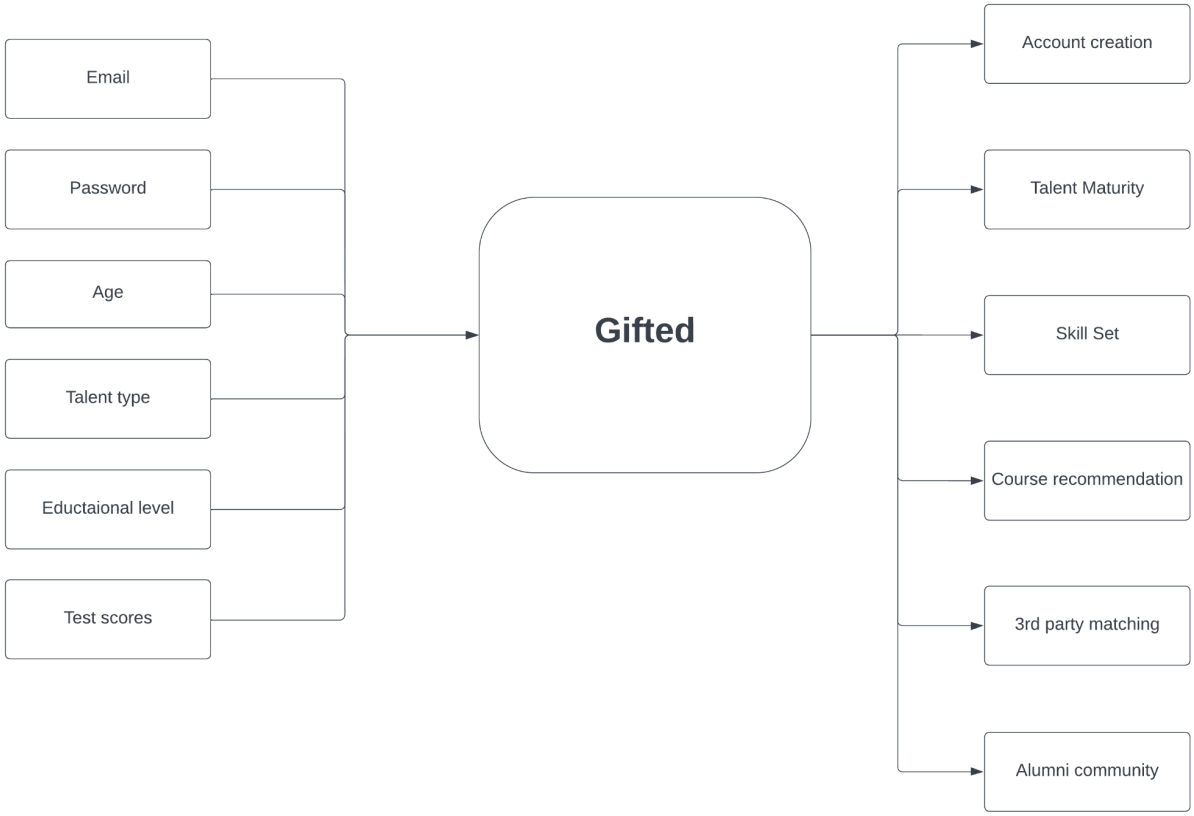


Fig 6.1 Abstract view diagram

6.2 Sub-system view

The sub-system diagram found below in Figure 6.2 represents all the components that our system (Gifted) consists of both internal and external sub-systems. External subsystems represent the components we will use directly without having to build them from scratch ourselves, yet, they are customizable enough for us to fit them to our use. Internal sub-systems on the other hand will be built by us from scratch, some pre-built code will be used, such as libraries for example.

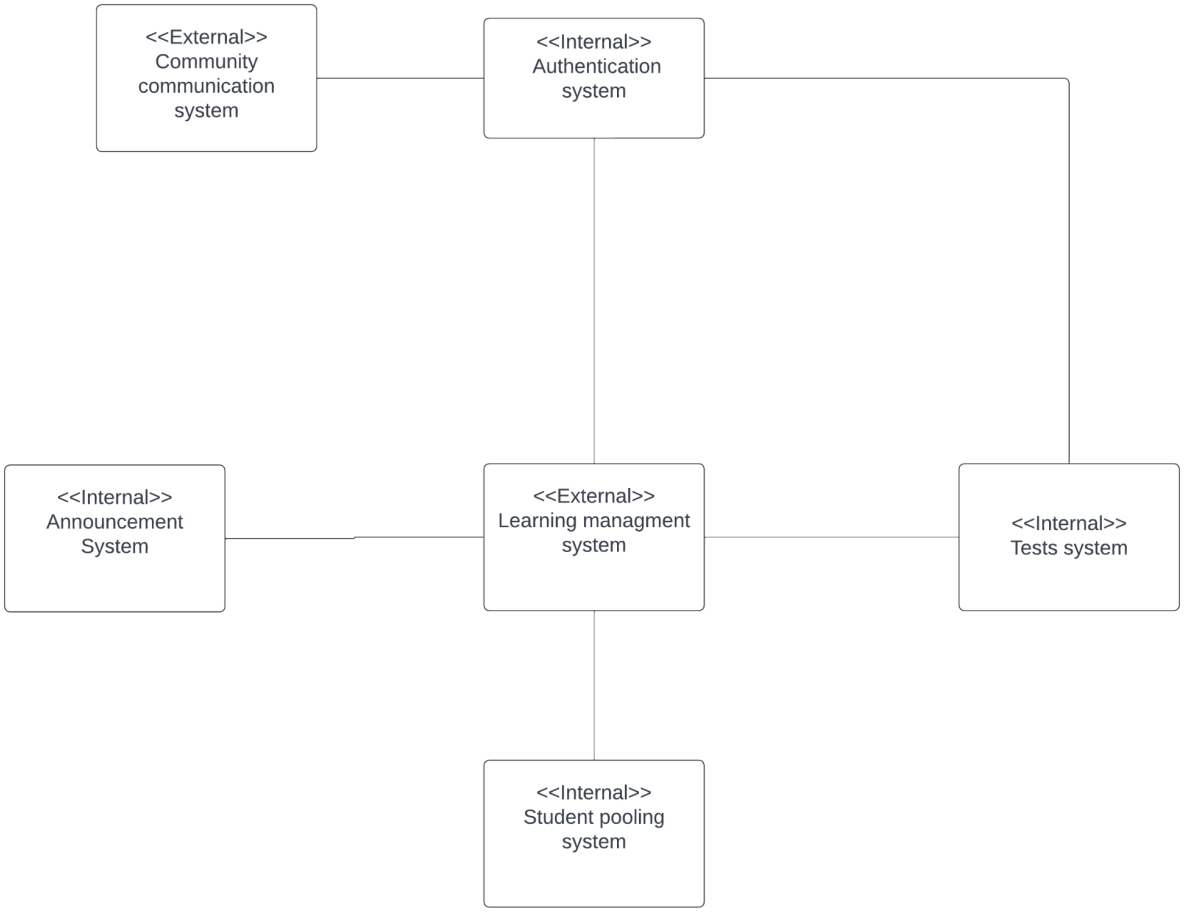


Fig 6.2 Sub-system diagram

7 System requirements

7.1 Functional requirements

The section covers the functional requirements of the system, functional requirements cover the functionality of the system, and what the system offers.

User shall be able to login into the system.

User shall be able to register his account in the system.

User shall be able to retrieve his password

User shall be able to change his password

Talent shall be able to participate in the pooling system

3rd party organizations user shall be able to view pools of talents

Talent shall be able to leave the pooling system

Admin shall be able to create an announcement

Admin shall be able to delete an announcement

Admin shall be able to create a test

Admin shall be able to delete a test

Talent shall be able to take a test

Talent shall be able to view test results

3rd party organizations users shall be able filter talents.

7.2 Non-Functional Requirements

7.2.1 Usability

User shall be able to learn the system within 20 minutes.
Talent shall be able to take test within 5 clicks

7.2.2 Reliability

System should not be in a failure state for more than 20 minutes per week.

7.2.3 Security

System should not allow talents to view other talents' grades.
System must be available and behave reliably even under DOS attacks.

7.2.4 Availability

System should be available 99.9% of the time.

7.2.5 Performance

System shall handle 500 users at the same time.
System shall be able to response no more than 50ms.

7.2.6 Maintainability

System Should not exceed more than 5 hours to fix the system.

8 System Analysis

8.1 Use case diagrams

8.1.1 Take test

Use Case Description
System: Gifted

Use Case name: Take test	
Primary actor: Talent	**Secondary actor(s):** None.
Description: This use case describes when a talent tries to take a test on the Gifted System	
Relationships **Includes:** View grade **Extends:** None	
Pre-conditions: 1. Talent must be signed in.	

Steps:		
Primary Actor	**System**	**Secondary Actor(s) (if applicable)**
1- Talent selects a test to take 3- Talent answers questions 4- Talent submits answers	2- System provides questions for talent 5- System shows the user their grade for the test 6-Save grade to database	

Alternative and exceptional flows: 1. a: Talent doesn't find a test Talent doesn't take a test and use cased ends with a failure 4.a: Talent quits: the use case ends with a failure condition
Post-Conditions: Successful condition: Grade for the user is saved Failure condition: Test grade is not saved

Table 8.1 Take test Use case diagram.

8.1.2 Enroll in a pool

Use Case Description	
System: Gifted	
Use Case name: Enroll in a pool .	
Primary actor: Talent	**Secondary actor(s):** None.
Description: This use describes how the user can enroll in a pool.	
Relationships **Includes:** None. **Extends:** Leave pool.	
Pre-conditions: 1. Talent must be signed in. 2. Talent must have passed the test.	
Steps:	

Primary Actor	**System**	**Secondary Actor(s) (if applicable)**

1. This use case begins when the Talent selects “Enroll in a pool” option. 3. The Talent chooses a pool’s field type. 5. The Talent chooses a pool.	2. The system displays the pool’s field types to choose from. 4. The system displays the available pools in that field to enroll in. 6. The system enrolls the Talent into the pool and lists it there based on their test score. 7. The system displays a message to the user indicating that the Talent has enrolled into the pool.	
Alternative and exceptional flows: 2.a. The Talent selects the ‘other’ option. 3.a. The system asks the Talent to enter the pool’s field type they want. 4.a. The Talent provides the pool’s field type. 5. The system displays a message indicating that a request has been made to create a pool in that field type.		
Post conditions: **Successful Conditions:** The Talent has enrolled into the pool or a request has been made to create a pool. **Failure Conditions:** The Talent did not enroll into the pool and no request has been provided..		

Table 8.2 Enroll in a pool Use case diagram.

8.1.3 Sign up

Use Case Description		
System: Gifted		
Use Case name: Sign up		
Primary actor: User	**Secondary actor(s):** None.	
Description: This use case describes when the user creates an account so that they can access the system.		
Relationships **Includes:** VerifyingEmail **Extends:** RecoveringCredentials		
Pre-conditions: None		
Steps:		
Primary Actor	**System**	**Secondary Actor(s) (if applicable)**
1-User clicks on “Sign up” button 2-User enters all required information including the email 5-User verifies his email from the email box.	3-System verifies the entered email. 4-System sends a verification email to the user to authenticate the talent. 6-System allows the user to access the website	

Alternative and exceptional flows: 2.a: User :enters false email 1-System displays a message indicating that the entered email is invalid 2.b: missing field: 1-System displays a message indicating that there is a missing field.
Post-Conditions: **Successful condition:** User is authenticated and allowed to access the system. **Failure condition:** User is not authenticated and access is not granted.

Table 8.3 Sign up Use case diagram.

8.1.4 Create announcement

Use Case Description		
System: Gifted		
Use Case name: Create announcement		
Primary actor: Admin	**Secondary actor(s):** None.	
Description: This use case describes when an admin tries to create an announcement.		
Relationships **Includes:** None **Extends:** Manage Announcements		
Pre-conditions: 1. Admin must be signed in.		
Steps:		
Primary Actor	**System**	**Secondary Actor(s) (if applicable)**

1- Admin selects create announcements. 3-Admin provides the information that is required including desired users.	2- System shows the announcement creation page. 4- System announces it to desired users.	
Alternative and exceptional flows: 3.a- Admin doesn't select users to receive the announcements.		
Post-conditions: **Successful conditions:** Announcement created successfully. Announcement published successfully. **Failure Conditions:** Announcement not published.		

Table 8.4 Create announcement Use case diagram.

8.1.5 Create test

Use Case Description		
System: Gifted		
Use Case name: Create test		
Primary actor: Admin	**Secondary actor(s):** None.	
Description: This use case describes when an admin tries to create a test		
Relationships **Includes:** None **Extends:** None		
Pre-conditions: 1. Admin must be signed in.		
Steps:		
Primary Actor	**System**	**Secondary Actor(s) (if applicable)**

1- Admin selects to create a Test. 3-Admin provides test questions and answers. 4-Admin provides weight for each question	2- System shows the Test creation page. 5- System register test in Database.	
Alternative and exceptional flows: 3.a- Admin doesn't provide answers to questions: Error is shown and admin is moved back to step 2. 4.a- Admin doesn't provide grades for questions: The total grade is equally divided between questions. 4.b- Total grades of questions is not 100 points: Error is shown and admin is moved back to step 2.		
Post-conditions: **Successful conditions:** Test created successfully. **Failure Conditions:** Test not created.		

Table 8.5 Create test Use case diagram.

8.2 Sequence diagram

Sequence diagrams showcase how the system behaves in execution time, the message communication sequence is shown in the diagram (Hence the naming). The following five diagrams cover the system across different subsystems.

8.2.1 Take test

This sequence diagram is concerned with the create test function, the user uses selectTest function and the boundary invokes takeTest and the entity should record the attempt in the system, the user should input the answers for the question which should update the attempt with the grade as shown in Figure 8.1.

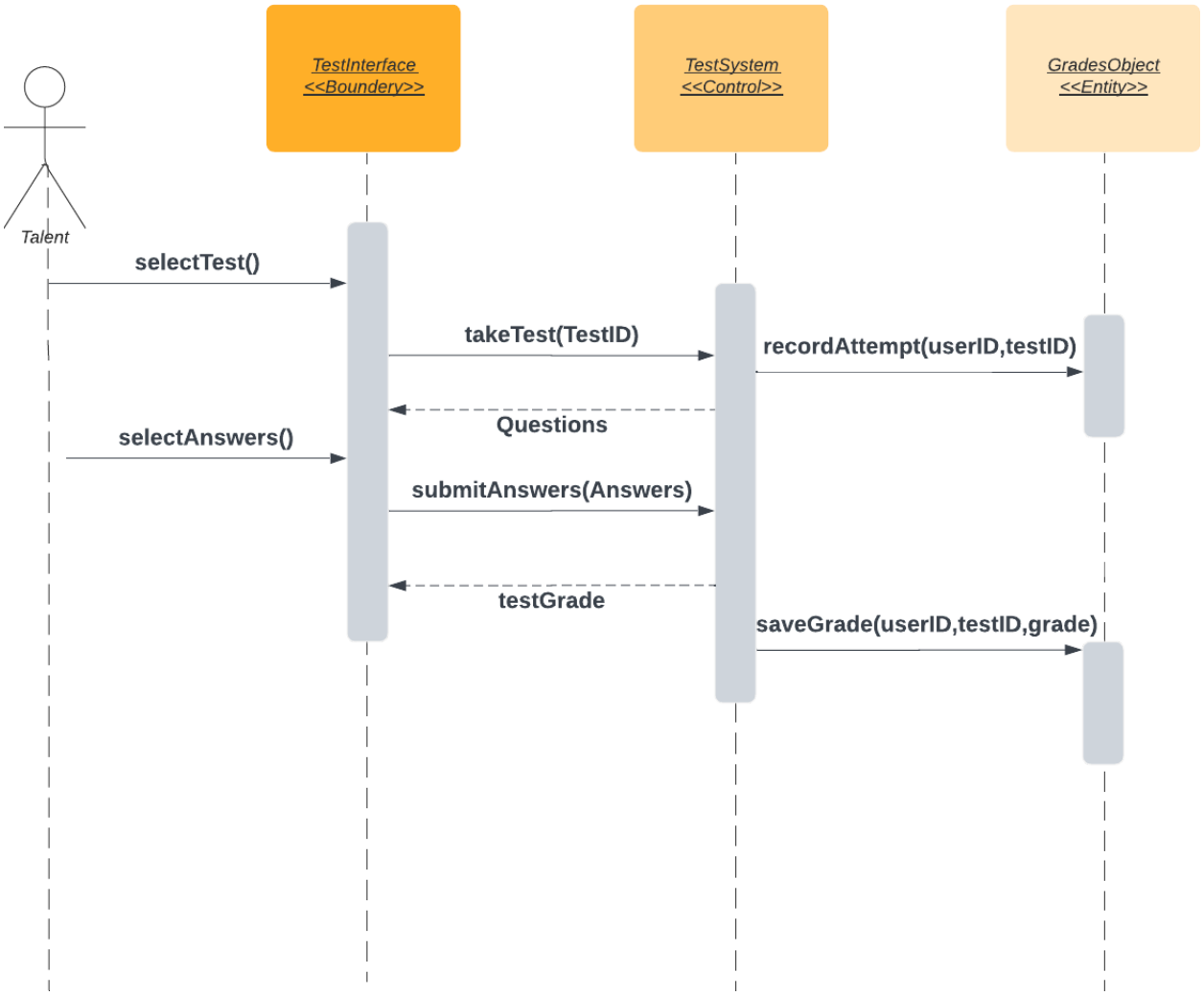


Fig 8.1 Take test sequence diagram

8.2.2 Enroll in a pool

The following sequence diagram showcases how the talent can enroll in a pool, the talent must first decide to enroll in the pooling system, then decide a pooling category using getPoolingCategories as shown in figure 8.2.

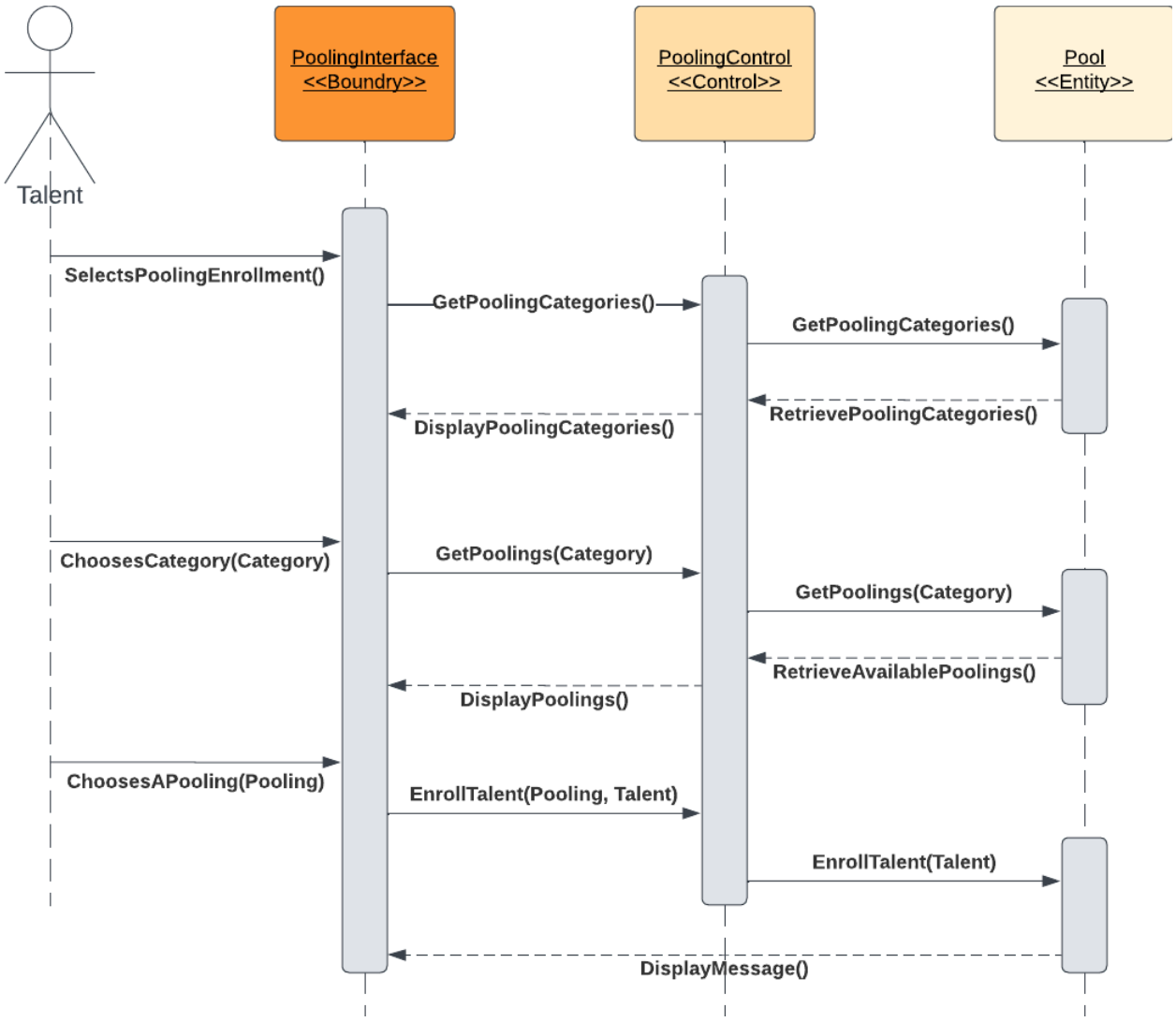


Fig 8.2 Enroll in a pool sequence diagram

8.2.3 Sign up

The signup sequence diagram is used to understand how a user can create an account in the system so that they can access the system features, the user must select signup, they provide their information. And the system should save the user's information in the entity as shown in the figure 8.3

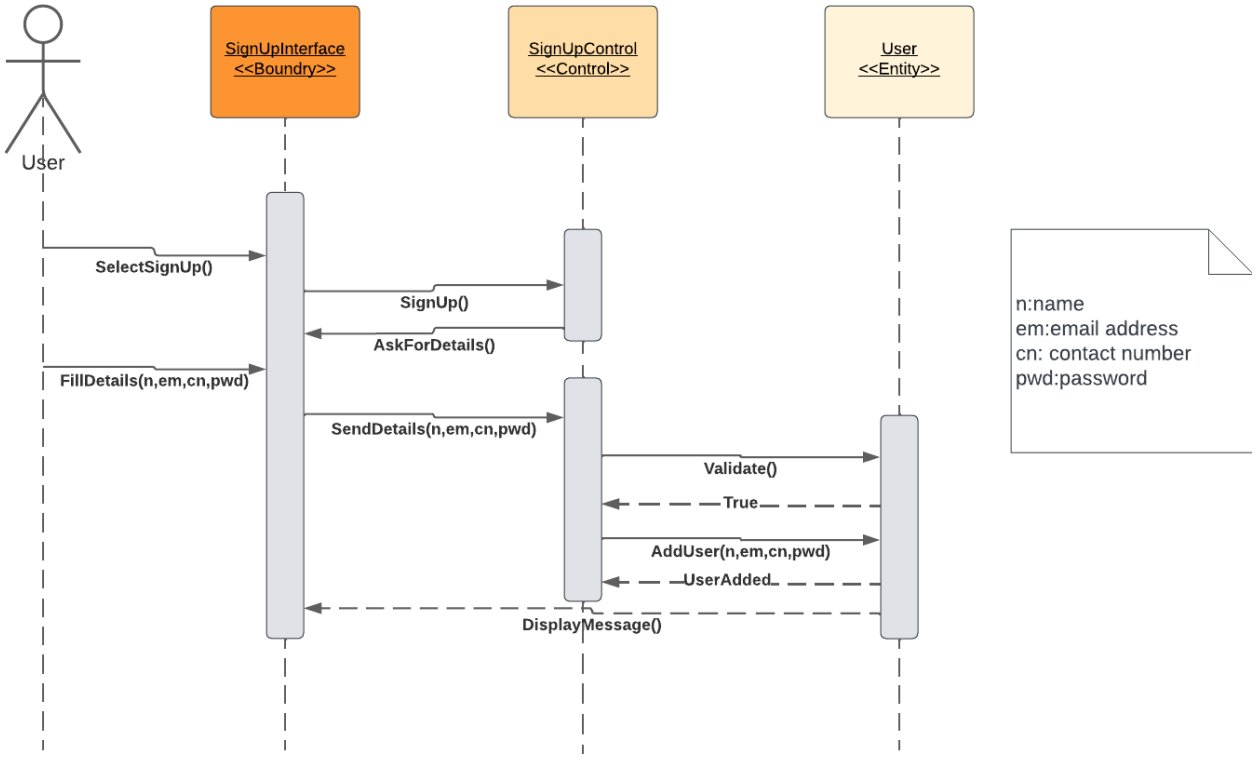


Fig 8.3 Signup sequence diagram

8.2.4 Create announcement

Figure 8.4 showcases the flow of events to create an announcement for the talents, an announcement is a message broadcasted to all the talents.

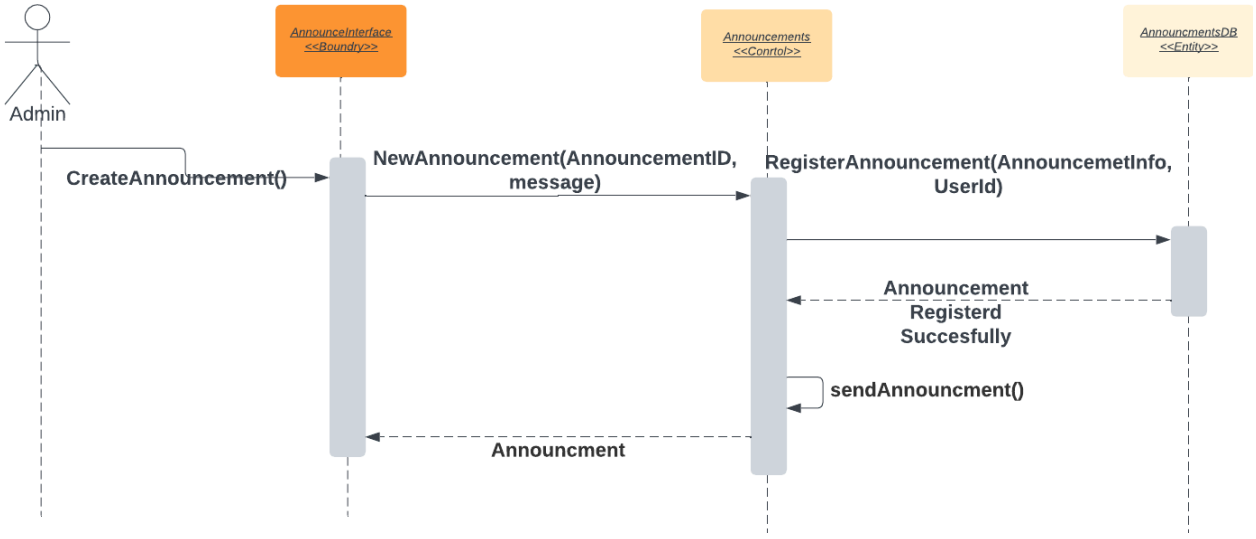


Fig 8.4 Create announcement Sequence Diagram

8.2.5 Create test

The figure 8.5 showcases the sequence diagram representing the flow at runtime that enables the admin to create a test for the talents.

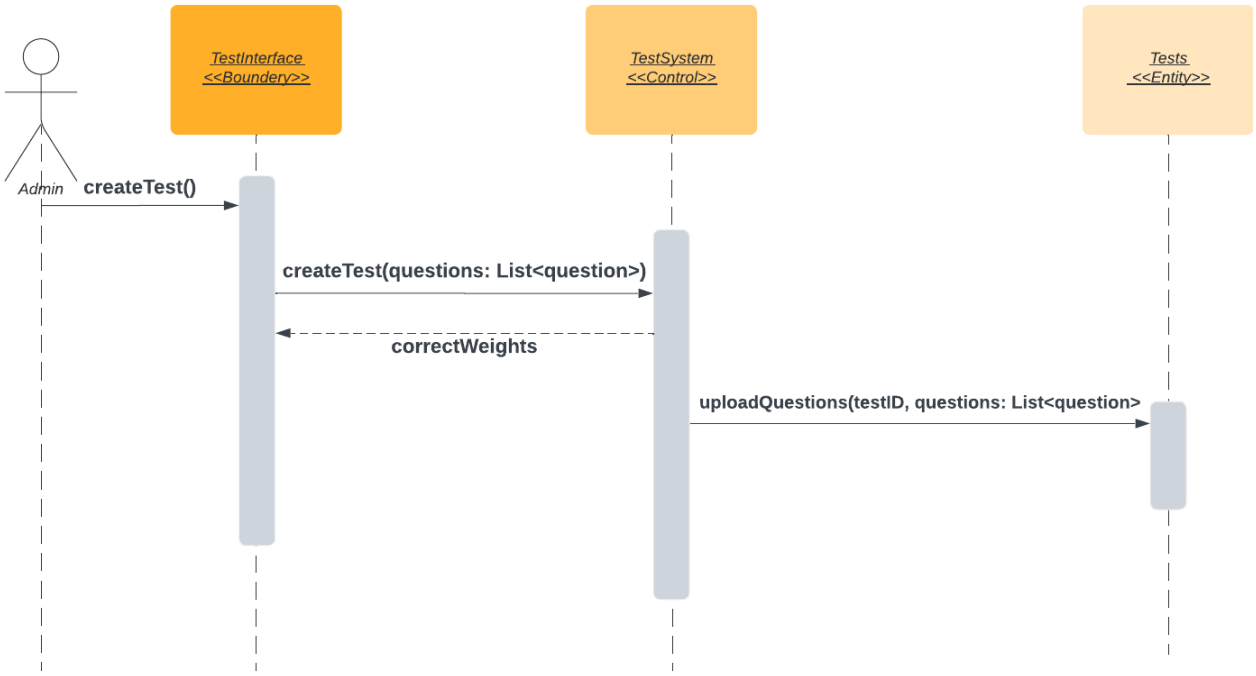


Fig 8.5 Create test sequence diagram

8.3 View of participating classes

VOPC diagram is concerned with all the classes used in the sequence diagrams in section 8.2, each diagram in this section corresponds to a sequence diagram.

8.3.1 Take test

The following figure diagram 8.6 showcases the participating classes in the take test function. The diagram consists of three classes, TestInterface, TestSystem, GradesObject.

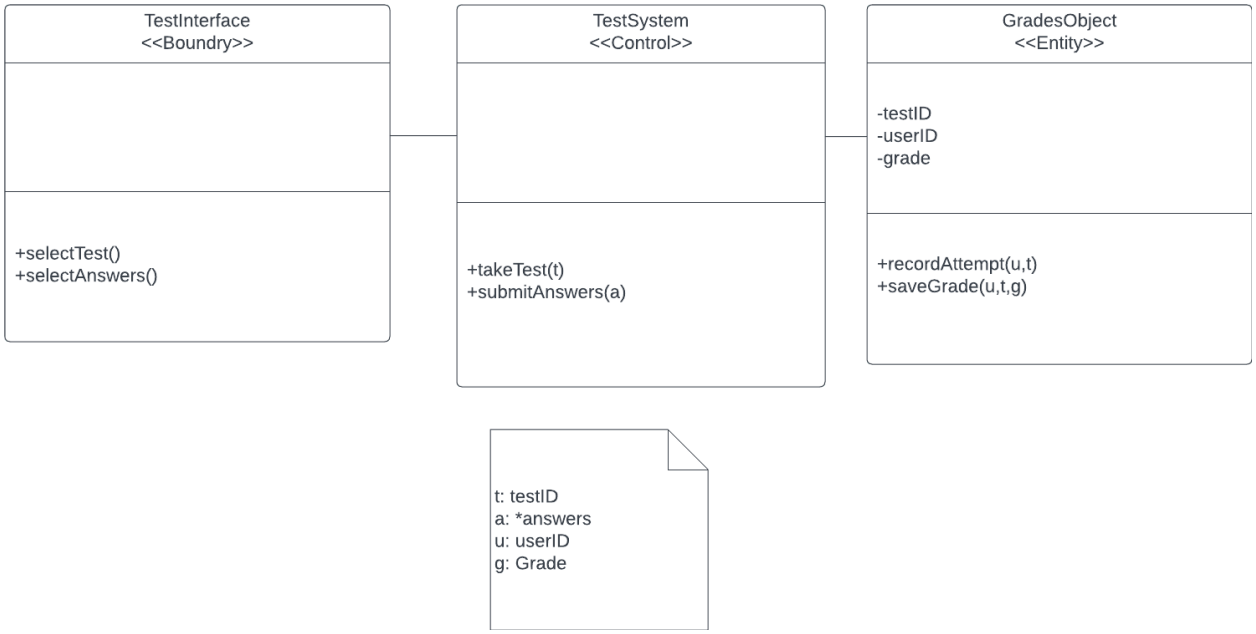


Fig 8.6 Take test VOPC Diagram

8.3.2 Enroll in a pool

The figure 8.7 showcases the classes used in the enroll in a pool function, which consists of 3 classes, PoolingInterface, PoolingControl and Pool.

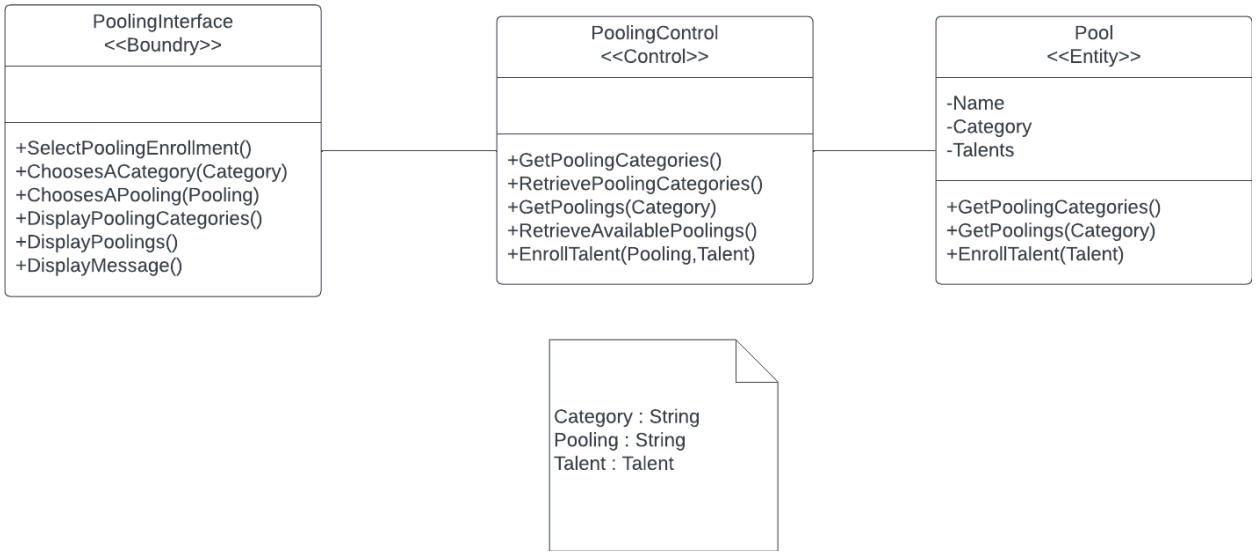


Fig 8.7 Enroll in a pool VOPC Diagram

8.3.3 Sign up

The following VOPC diagram Figure 8.8 showcases the classes participating in the signup function in the system, which consists of, Signup Interface, SignupControl and User.

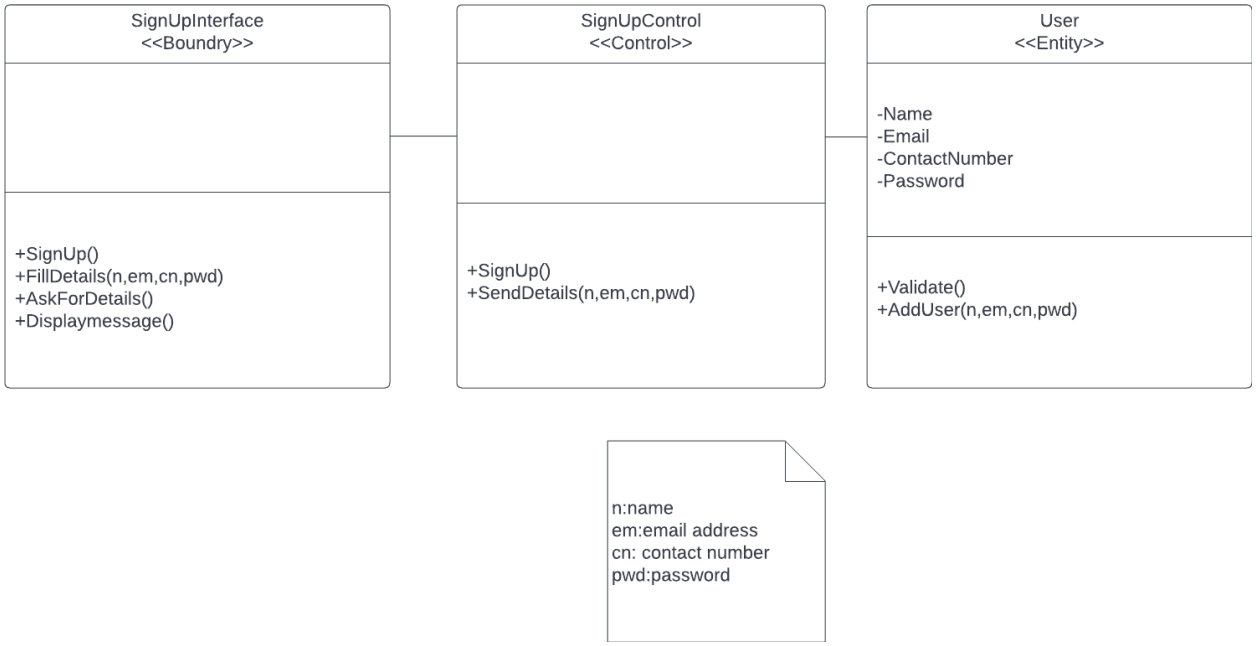


Fig 8.8 Sign up VOPC Diagram

8.3.4 Create announcement

The following Figure 8.9 showcases the diagrams involved in the announcement creation function, which are, AnnouncementInterface and AnnouncementDB.

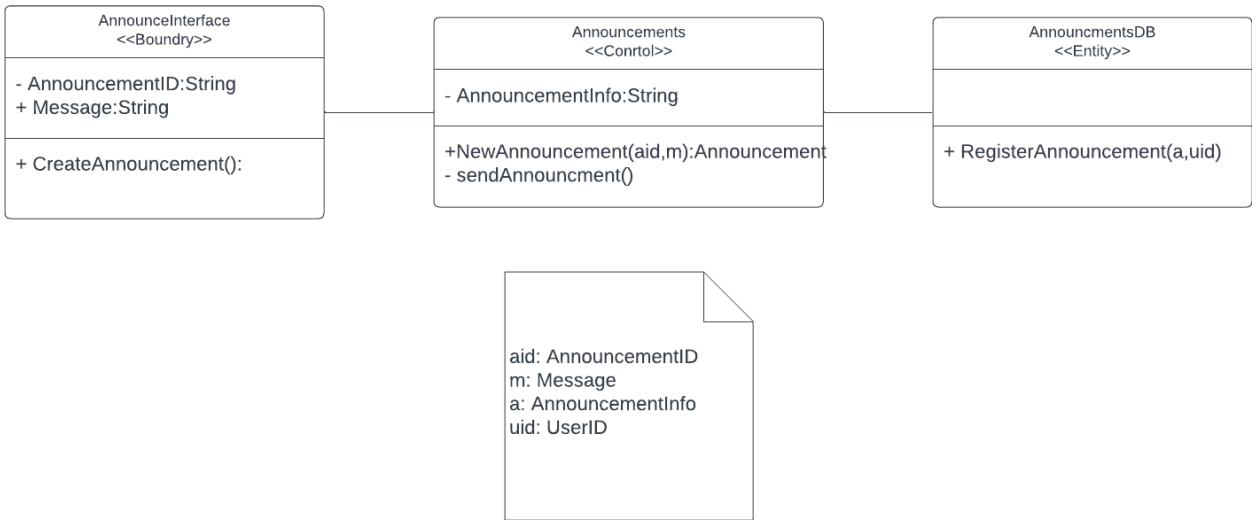


Fig 8.9 Create Announcement VOPC Diagram

8.3.5 Create test

Figure 8.10 Showcases the participating classes in the create test function, which allows admins to create tests. The classes are TestInterface, TestSystem, Tests.

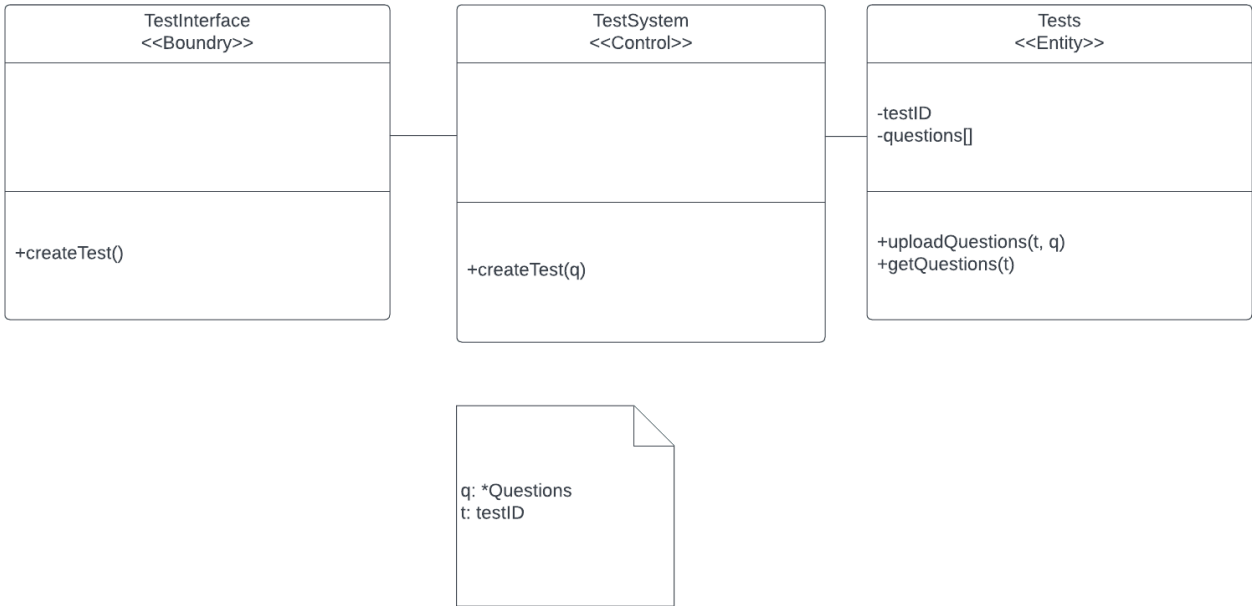


Fig 8.10 Create test VOPC Diagram

8.4 Class design of the system

Figure 8.11 shows the VOPC diagram of the whole system, it combines all the figures above and the relationships between them.

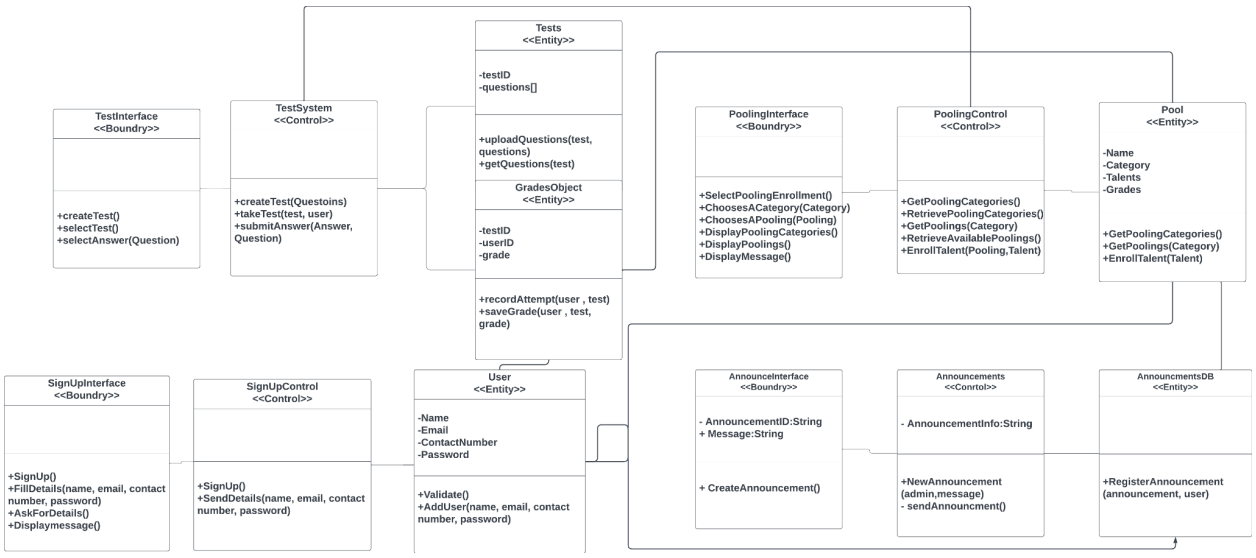


Fig 8.11 VOPC of system

9 System architecture

9.1 System design style

The system is considered a web application with entities, boundaries, and a controller to manipulate data by providing and updating data to the data store. This is considered best dealt with using the MVC architecture. Also, Turning a complex application development process into a much more manageable one is the main reason for choosing the MVC architecture. It helps us manage the frontend and backend in smaller separate components allowing for the application to be scalable, maintainable, and easy to expand. As seen in figure 9.1.

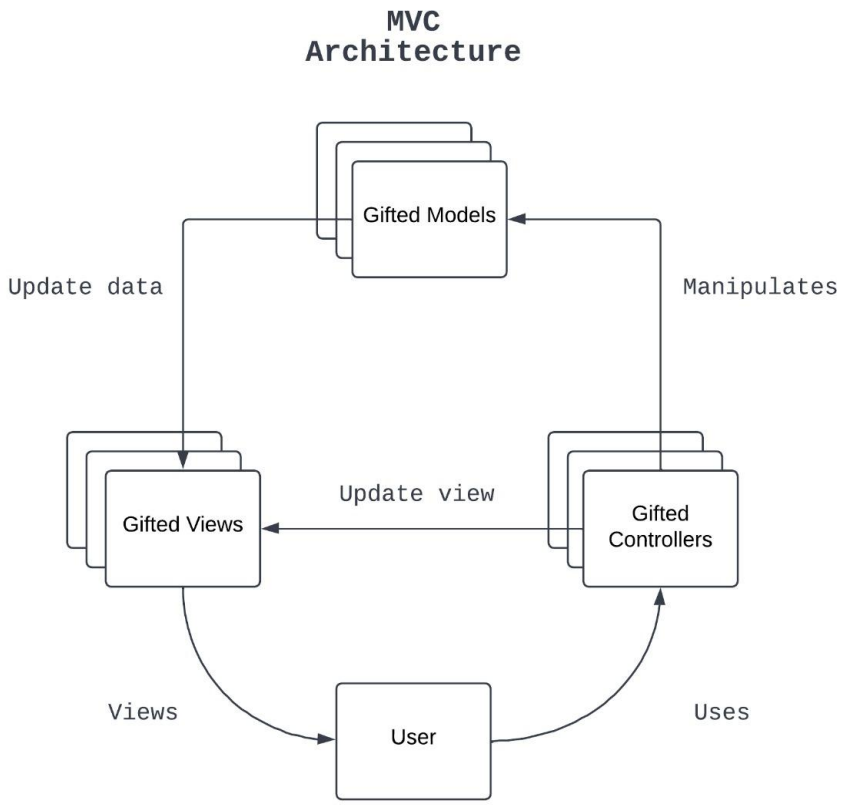


Fig 9.1 Architecture design style

9.2 System data center design

For storing the data and retrieving it, we considered more than one data center design style, after reviewing our requirements and the system needs, we landed on a data centered repository style. The figure 9.2 represents how the system will store and interact with the data repository.

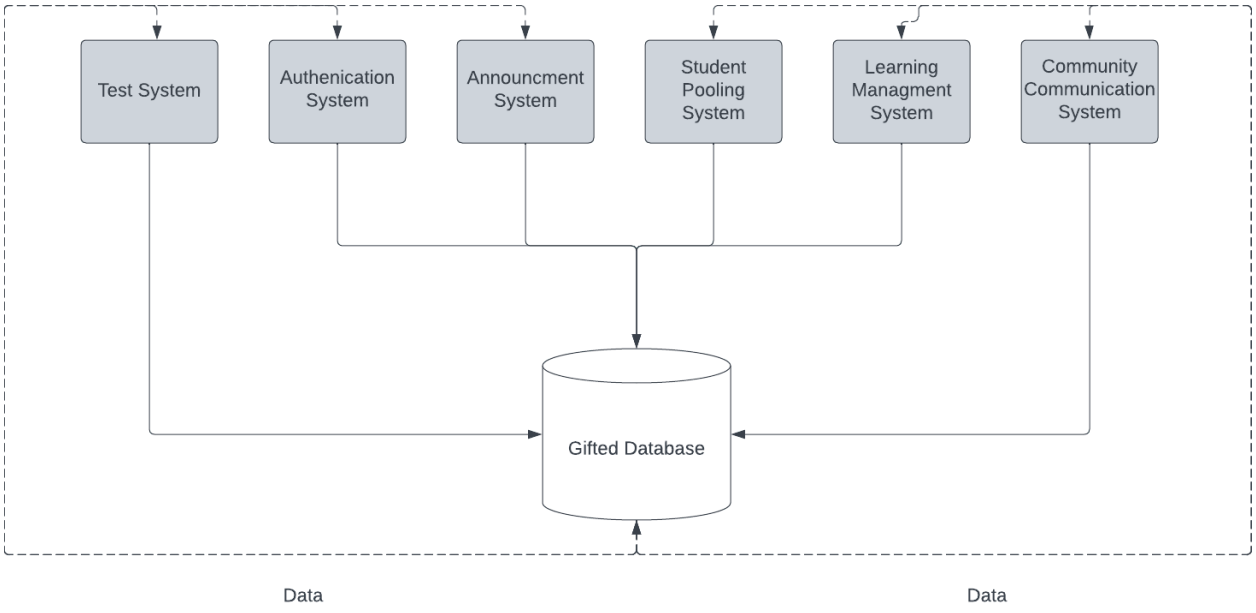


Fig 9.2 Repository data center

9.3 ER diagram

This is a description of data aspects of a business domain to our system. That contains entities and the relationships that can exist among them, and databases. As shown in Figure 9.3

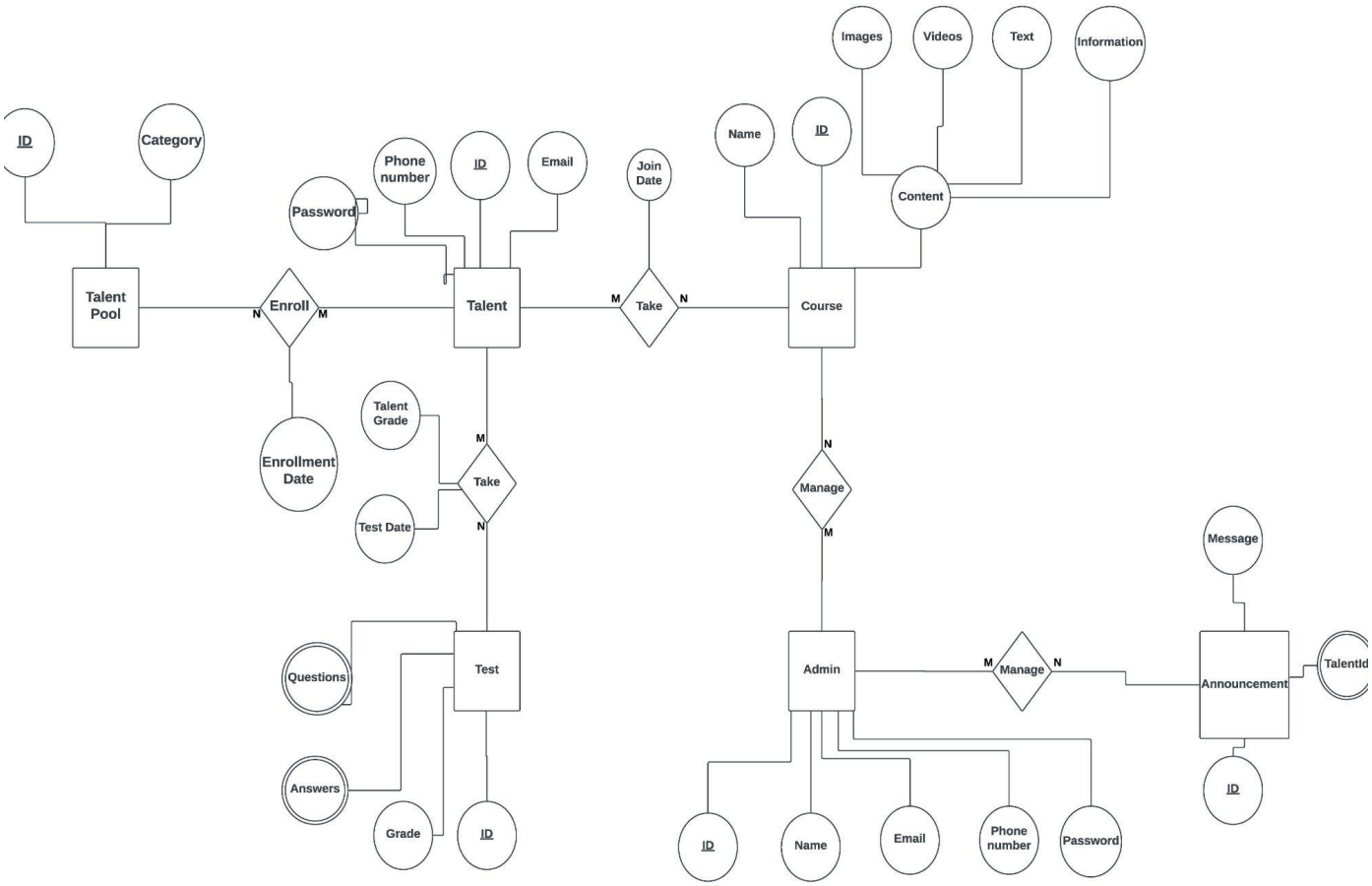


Fig 9.3 ER diagram

10 Testing the system

We want to make sure every functionality of the system is developed to exactly do as required, such that we can achieve our goal of building a system that genuinely helps talents grow. The table below showcases the tests that must be performed on the system. All these tests are set before the system development, so we can't conclude the test result yet.

Test Case	***Scenario***	***Input***	***Expected output***
Take Test	The user selects a test to take, answers the questions and submits their answers	Answers to test questions	Test answers are submitted and grade is shown
View tests	User goes to the tests page and views all the available tests	Test category	All tests are shown

Make announcement	Admin makes an announcement and sends it to all available students	Announcement text	Students receive and announcement with the text sent by the admin
talent enroll in a pool	talent goes to their settings and decides to enroll in a pool	Contact information	User is shown in talent pool
Create Test	Admin creates a test for the talents	Test name, questions and expected answers	Test is shown to students
Delete Test	Admin selects a test and chooses delete	Test to be deleted	Test shouldn't be available to talents but the grade should still be available to previous talents
Register	User registers to use the system	Email, password	An account should be created for the user
Login	User logins to their account	Email, password	User should be able to access the system

Table 10.1 Testing the system.

11 Algorithms

The system uses multiple algorithms developed by the system developers, the following pseudocode shows the most important algorithms developed.

11.1 Take a Test

A test options will be shown to the user and from them the user will choose one.

```
selectTest(){
        READ INPUT and SAVE in ChosenTest
        CALL takeTest(ChosenTest, User)
}

takeTest(ChosenTest, User){
        DOUBLE grade
        FOR EACH Question{
                OUTPUT Question FROM database
                CALL selectAnswer(questionID)
```

```
            grade = grade + GET grade from DATABASE
        }
        CALL recordAttempt(ChosenTest, User)
        CALL saveGrade(ChosenTest, User, grade)
}

selectAnswer(questionID){
        READ INPUT and SAVE in Answer
        CALL submitAnswer(Answer, questionID)
}
```

11.2 Create Announcement

This code runs after the Admin presses the create button.

```
createAnnouncement(){
        READ INPUT and SAVE in Message
        CALL newAnnouncement(Message, Admin)
}

registerAnnouncement(Message, Admin){
        Create Announcement in DATABASE
        ADD Message in DATABASE
        IF Announcement created successfully
            displayAnnouncement()
        ELSE
            OUTPUT(“Something went wrong”)
}
```

11.3 Enter A Pool

This code will run after the Talent chooses a category from the list.

```
choosesACategory(){
        READ INPUT and SAVE in category
        CALL getPoolings(category)
}

getPoolings(category){
        GET poolings in this category FROM DATABASE
        CALL displayPoolings()
}

choosesAPooling(){
        READ INPUT and SAVE in pool
```

```
        CALL enrollTalent(Pool, Talent)
}
enrollTalent(Pool, Talent){
        ADD Talent to Pool in DATABASE
        IF added successfully
                OUTPUT("You have been enrolled in the pool")
                displayPool()
        ELSE
                OUTPUT("Something went wrong , try again later")
}
```

12 User Interface Mockups

The system has multiple interfaces for the users to interact with, these Mockups below demonstrate how the system will appear for the end users.

12.1 User login

Figure 12.1 shows the Mockup for the login page.

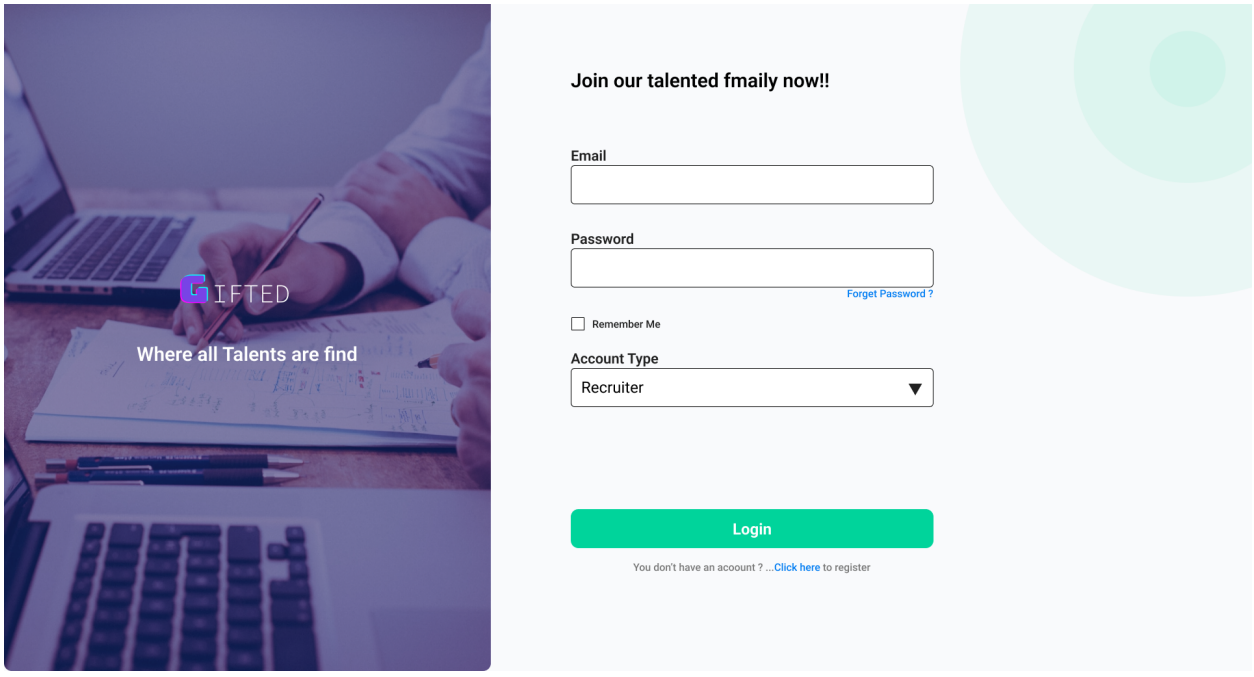


Fig 12.1 Login page Mockup

12.1 Talent Homepage

Figure 12.2 shows the Mockup for the talent homepage.

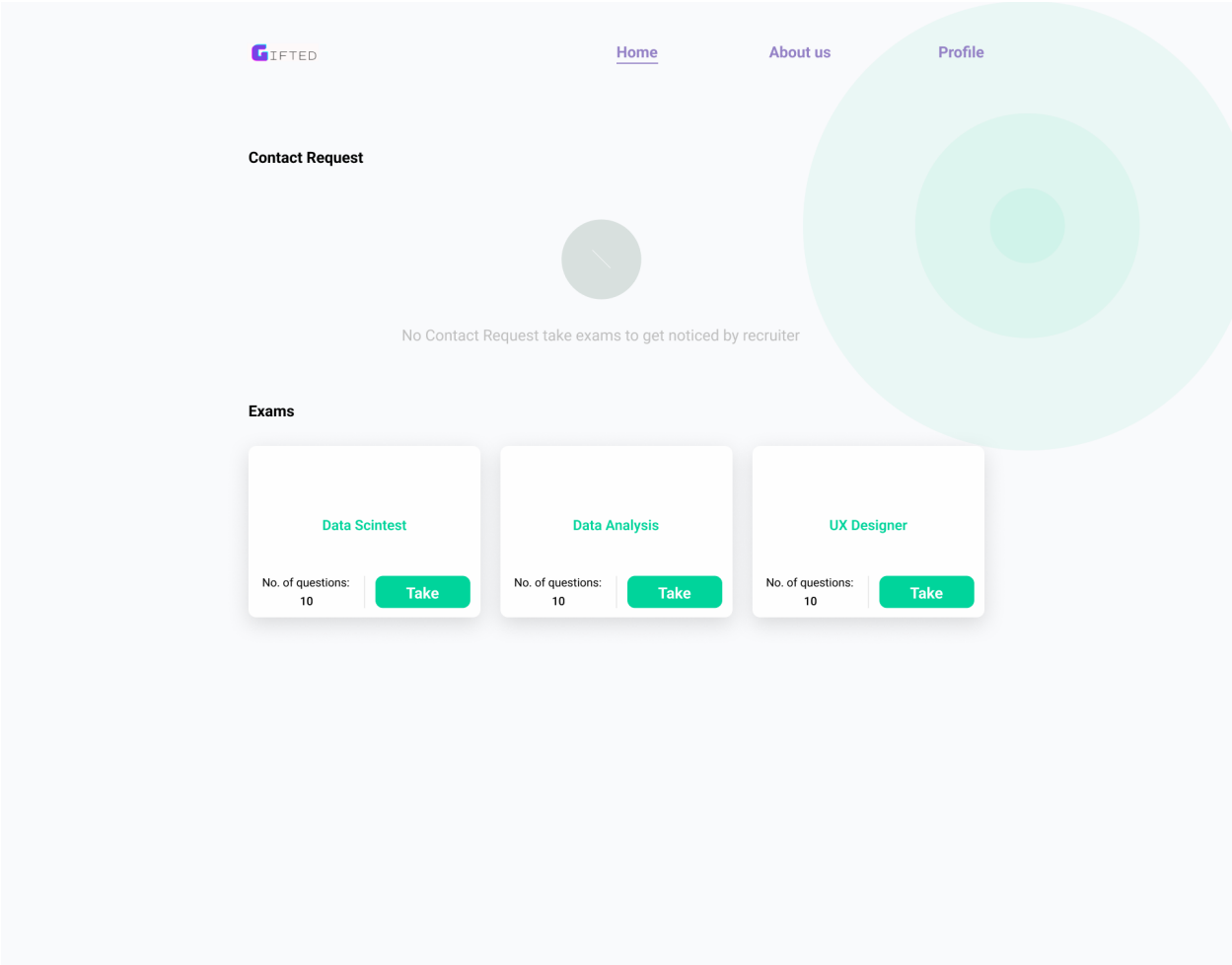


Fig 12.2 Talent homepage Mockup

12.3 Talent exam

If a talent chooses to enroll in a pool of talents for a certain category, he will take an exam related to that category, Figure 12.3 shows an example of that exam.

GIFTED

Home About us Profile

Data Analysis enrollment exam

Q1 First Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q2 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q3 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q4 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q5 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q6 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q7 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q8 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q9 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Q10 Question ?

- (•) Answer 1
- ( ) Answer 2
- ( ) Answer 3
- ( ) Answer 4

Submit

Fig 12.3 Talent exam Mockup

12.4 Pool of talents

Figure 12.4 shows the Mockup for a certain categorized pool of talents.

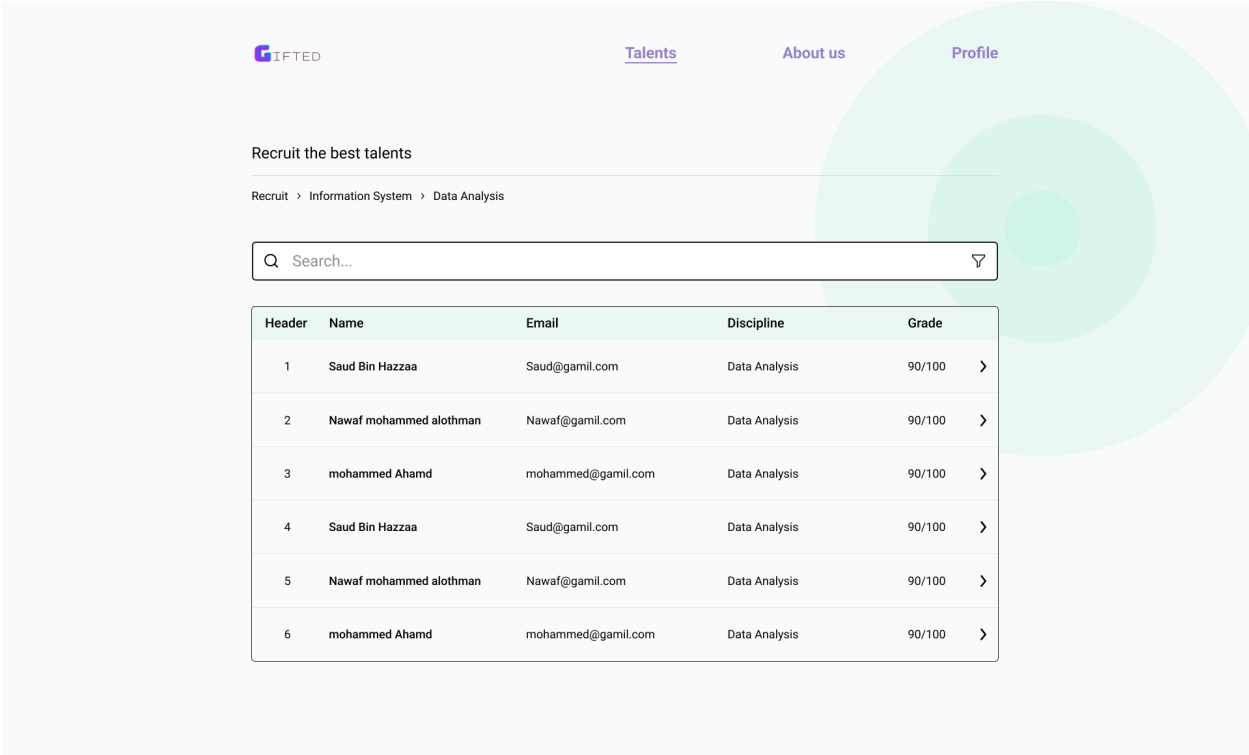


Fig 12.4 Talents pool Mockup

12.5 Talent profile page

After viewing a certain talent pool, third-party organizations can view the profile of a talent they are interested in contacting, or simply if a talent wants to view his profile to add contact info and see his exam grades, Figure 12.5 demonstrates the talent profile Mockup.

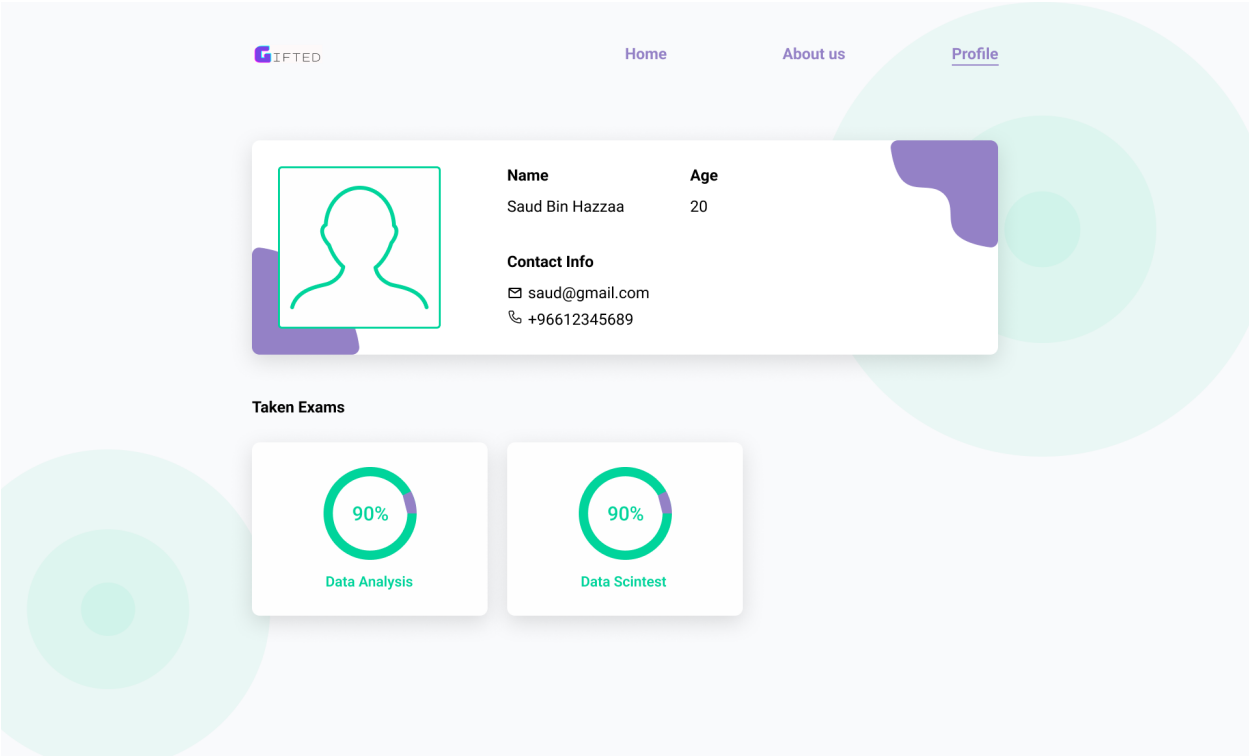


Fig 12.5 Talent profile Mockup

13 Project Status

Building the system is no easy task, but we believe that a major part is finished as we finish this document. The analysis and all pre-implementation analysis required is done. The project can move to the implementation phase. Some issues were faced regarding the scope of the project, as our future goal is to build an ecosystem to help talents. But, we believe this is a good start.

14 Conclusion

In conclusion, the system and our understanding of how we can help talents the most we can has improved greatly. We know how the system is going to be developed using the system architecture, sequence diagrams, and VOPC. We know what the system is going to do using use case diagrams, and functional requirements. We know how the system will be tested using the test cases developed. We developed the algorithms we will need to build the system. And we know how to deploy the system using the deployment diagram.

For the next phase of the project, we will be working on implementing Gifted.This document explains in detail how to implement the system, which will greatly accelerate the progress of the next phase, and most importantly improve the quality of the project.

References

1. Scalers, www.withscalers.com, 2019
2. Talented Athlete Scholarship Scheme, www.tass.gov.uk, 2004
3. Qimam, www.qimam.com/en, 2018
4. Institute for Educational Advancement, www.educationaladvancement.org, 1998
5. Mawhiba, www.mawhiba.org/Ar/Pages/default.aspx, 1999
6. Johns Hopkins Centre for Talented Youth, www.cty.jhu.edu, 1979
7. 1stalent, www.1stalent.com/en, 2009
8. Markoding, www.markoding.org, 2017