

# Tripple A

## Project Documentation Template

Prepared by:

**Ahmed Reda Ewis**

**Ahmed Mohamed Hamdy**

**Muhammad Ahmed Kadry (Group Leader)**

**Zeyad Khaled Mohamed**

**Tarek Nasr Tayel**

**Track**

**Professional Web Development & Bi**

**Project Supervisor**

**Eng: Momen Khamis**

**Intake 44**

**2024**

1. Introduction .....	3
1.1 BACKGROUND(PROBLEM).....	
1.2 PURPOSE.....	
1.3 PREVIOUS WORK DONE(COMPETITORS ANALYSIS).....	
1.4 CUSTOMERS' ANALYSIS.....	
1.5 SCOPE.....	
1.6STAKEHOLDERS/BENEFICIARIES.....	
1.7 BUSINESS MODEL.....	
2. Requirements.....	8
2.1 <i>USER/FUNCTIONAL REQUIREMENTS</i> .....	
2.1.1<List of Functional Requirements or Features>.....	
2.2 USE CASES.....	
2.2.1 Actors.....	
2.2.2 List of Use Cases.....	
2.2.3 <i>Use Case Diagrams</i> .....	
2.3 CLASSES	
2.3.1 <List of Classes >.....	
2.4 NON-FUNCTIONAL REQUIREMENTS.....	
2.4.1 Performance Requirements(Availability, Security .....etc.).....	
3. Objectives/List of Services(measurable) .....	21
4. Design Overview .....	25
4.1 SYSTEM ARCHITECTURE.....	
4.2 DATA DESIGN(ENTITY RELATIONSHIP DIAGRAM) .....	
5. Implementation .....	30
5.1 <i>TOOLS &amp; TECHNOLOGIES</i> .....	
5.2 <i>HARDWARE REQUIREMENTS</i> .....	
5.3 <i>STEPS OF INSTALLATION</i> .....	
6. User/Customer Testing.....	35
6.1 <i>USERS/CUSTOMERS FEEDBACK</i> .....	
6.2 <i>Improvements done</i> .....	
7. Recommendations(Future Work).....	37
8. User Guide(See Attachment).....	38
8.1 <i>QUICK START GUIDE</i> .....	
8.2 <i>MAIN SCENARIOS</i> .....	
8.3 <i>TROUBLESHOOTING</i> .....	

# 1. Introduction

## 1.1 Background(Problem)

In the age of rapid technological advancements, the need for accurate and timely information has become more crucial than ever. People, especially those in technical fields, frequently encounter complex problems that require immediate solutions. Traditional methods of seeking help, such as consulting books or contacting experts, often prove to be time-consuming and inefficient. This has given rise to the popularity of online question-and-answer platforms, where users can seek advice and solutions from a global community of experts and enthusiasts. However, many existing platforms have limitations such as poor user experience, unstructured content, and inadequate moderation, which lead to issues like misinformation, redundant questions, and low-quality answers.

## 1.2 Purpose

The purpose of this project is to create a user-friendly, efficient, and reliable question-answering site that addresses the shortcomings of existing platforms. This site will enable users to post questions, provide answers, and engage in discussions, ensuring that the information shared is accurate, high-quality, and easily accessible. By implementing advanced search functionalities, user reputation systems, and robust moderation tools, the platform aims to foster a collaborative and supportive community where users can efficiently solve their problems and share knowledge.

## 1.3 Previous work done(Competitors Analysis)

Several prominent question-answering platforms have set a benchmark in this domain. Sites like Stack Overflow, Quora, and Reddit have large user bases and a wealth of content. However, each of these platforms has its own set of strengths and weaknesses:

- **Stack Overflow**

**Strengths:**

- Comprehensive Tagging System: Allows users to categorize questions effectively, making it easier to find relevant information.
- Detailed Guidelines: Provides clear guidelines for asking and answering questions, which helps maintain content quality.

**Weaknesses:**

- Intimidating for New Users: Strict moderation and community rules can discourage new users from participating.
- Limited Scope: Primarily focused on programming and technical questions, which may not appeal to users seeking information on other topics.
- High Entry Barrier: Users with low reputation might find it difficult to participate fully, such as by adding comments or voting.

- **Quora**

**Strengths:**

- Wide Range of Topics: Covers a broad spectrum of subjects, from technical to personal to philosophical questions.
- User-Friendly Interface: Easy-to-navigate interface with features like personalized feed and notifications.

**Weakness:**

- Too much Ads : There are too much ads which may affect the user experience negatively.
- Promoted Questions: the promoted question aren't related to the question which the users search for which may be confusing .

• **Reddit**

**Strengths:**

- Engaging Community: Encourages in-depth discussions and diverse viewpoints, fostering a vibrant and interactive community.
- User-Friendly Interface: Easy-to-navigate interface with features like personalized feed and notifications.

**Weakness:**

- Trolling and Spam: Popular subreddits can attract trolls and spam, which can detract from the user experience.

## 1.4 Customers' Analysis

The primary customers of this platform are individuals seeking answers to technical and non-technical questions. These include:

Students and Researchers: Looking for academic assistance and research guidance.

Professionals and Developers: Seeking solutions to work-related problems or technical issues.

Enthusiasts and Hobbyists: Interested in learning and sharing knowledge about their hobbies or areas of interest.

General Public: Looking for answers to everyday problems and general inquiries.

All of them need a user-friendly platform without scam posts or over strict supervision.

## 1.5 Scope

The scope of this project includes the development, testing, and deployment of a question-answering site with the following features:

- User registration and authentication.
- Question posting and answering.
- Advanced search functionality.
- User reputation and reward system.
- Moderation tools to ensure content quality.
- Community guidelines and policies.
- Responsive design for mobile and desktop use.
- Data privacy and security measures.

The project will also involve continuous improvement based on user feedback and the addition of new features to enhance user engagement and satisfaction.

## 1.6 Stakeholders/Beneficiaries

The key stakeholders and beneficiaries of this project include: Users: Who will benefit from a reliable source of information and

community support. Moderators: Who will help maintain the quality and integrity of the platform. Developers: Who will gain experience and knowledge from building and maintaining the site. Advertisers and Sponsors: Who can reach a targeted audience through the platform.

## 1.7 Business Model

The business model for the platform will focus on generating revenue through multiple streams:

Advertisements: Displaying relevant ads to users based on their activity and interests.

Corporate Partnerships: Collaborating with companies and educational institutions to provide tailored solutions and services.

Donations and Sponsorships: Allowing users and organizations to support the platform financially.

By diversifying revenue streams, the platform aims to sustain its operations and continuously improve its services.

Our plan to attract the users to this platform based on persuading the experts to build their personal branding here instead of contributing to the highly competitive platforms and therefore the ordinary users will prefer user-friendly, free scam and tolerant but vigilant supervision platform

## 2. Requirements

### 2.1 User/Functional Requirements

This section details the core functionalities and features that the question-answering site must support to meet user needs and project objectives.

#### 2.1.1<List of Functional Requirements or Features>

##### User Registration and Authentication:

Users must be able to create accounts using email or third-party services (e.g., Google, Facebook). Secure login/logout functionality. Password recovery and reset options.

##### Profile Management:

Users can create and update their profiles, including adding a bio, profile picture, and personal information. Users can view their activity history, such as questions asked, answers provided, and reputation points.

##### Question Posting:

Users can post new questions with titles, detailed descriptions, tags, and optional media attachments. Question formatting options (e.g., markdown support).

##### Answer Submission:

Users can submit answers to questions. Answer formatting options similar to question posting.



### Voting and Reputation System:

Users can upvote or downvote questions and answers. Reputation points awarded based on upvotes and other contributions.

### Commenting:

Users can comment on questions and answers to seek clarification or provide additional information.

### Search and Filtering:

Advanced search functionality to find questions and answers using keywords, tags, and filters. Sorting options (e.g., most recent, most upvoted).

### Moderation Tools:

Admins and moderators can edit, delete, or flag inappropriate content. Tools for handling spam, abuse, and rule violations.

### Notification System:

Users receive notifications for various activities, such as new answers to their questions, upvotes, and comments.

### Community Guidelines and Help:

Accessible documentation on community guidelines, usage policies, and help resources.

## 2.2 Use Cases

This section describes the interactions between users and the system, detailing the different ways users can achieve specific goals using the platform.

## 2.2.1 Actors

**Registered User:** An authenticated user who can ask questions, provide answers, vote, and comment.

**Moderator:** A user with special permissions to manage content and enforce community guidelines.

## 2.2.2 List of Use Cases

- **User Registration:**

Actor: Guest User

Description: A guest user registers an account using email or a third-party service.

- **Login/Logout:**

Actor: Registered User

Description: A registered user logs into or out of their account.

- **Post a Question:**

Actor: Registered User

Description: A user posts a new question with details and tags.

- **Answer a Question:**

Actor: Registered User

Description: A user submits an answer to a posted question.

- **Vote on Content:**

Actor: Registered User

Description: A user upvotes or downvotes questions and answers.

- Comment on Content:

Actor: Registered User

Description: A user adds a comment to a question or answer.

- Search for Content:

Actor: Registered User

Description: A user searches for questions and answers using keywords and filters.

- Moderate Content:

Actor: Moderator

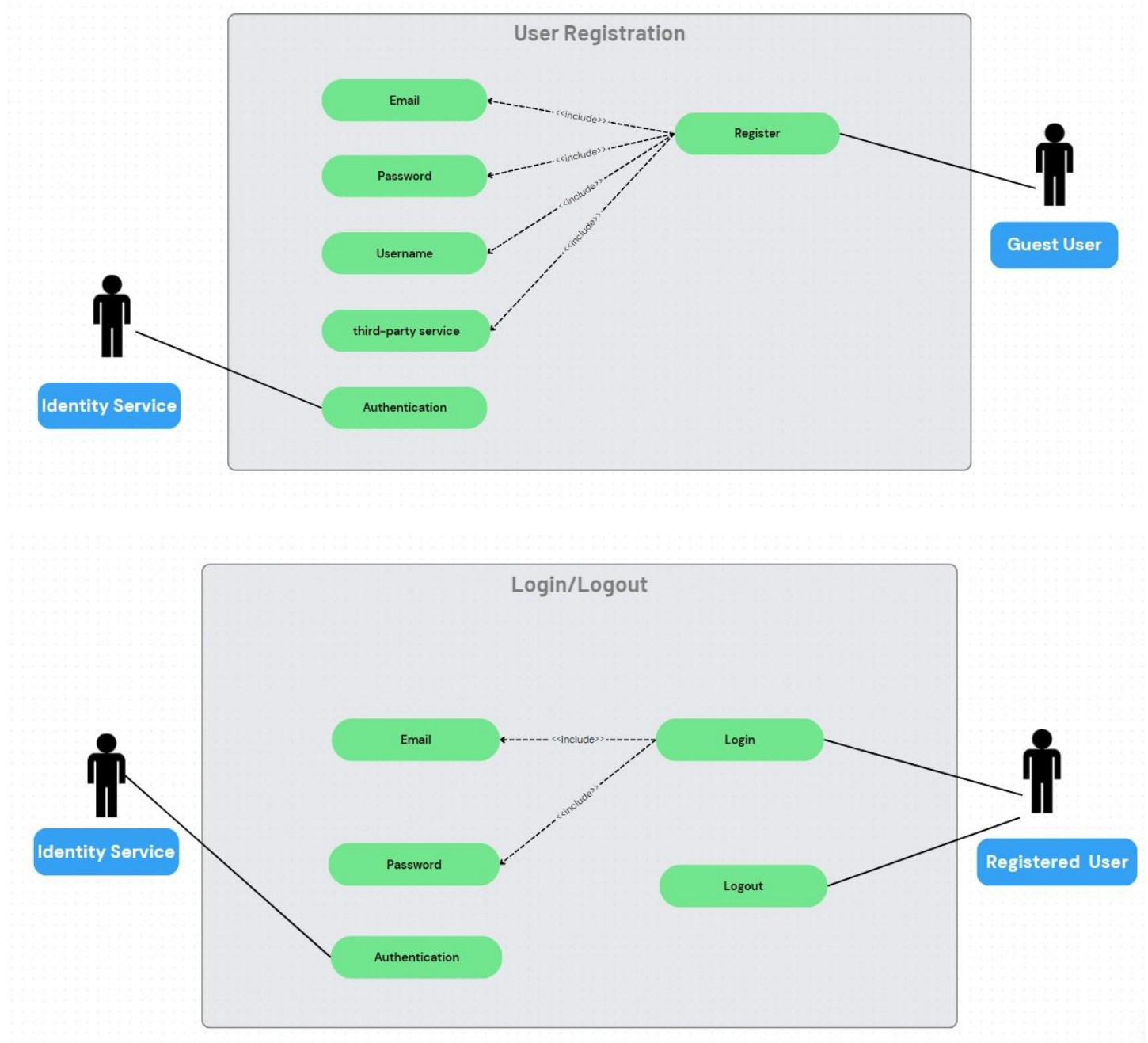
Description: A moderator edits, deletes, or flags inappropriate content.

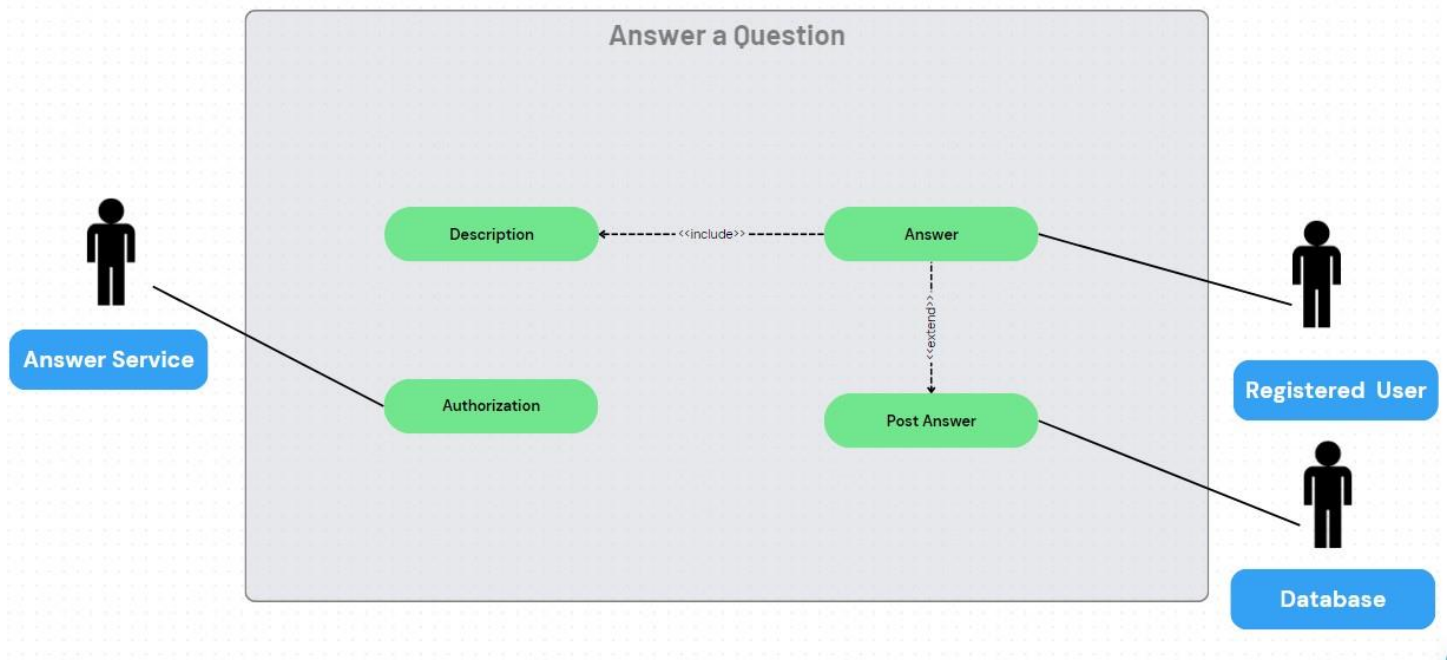
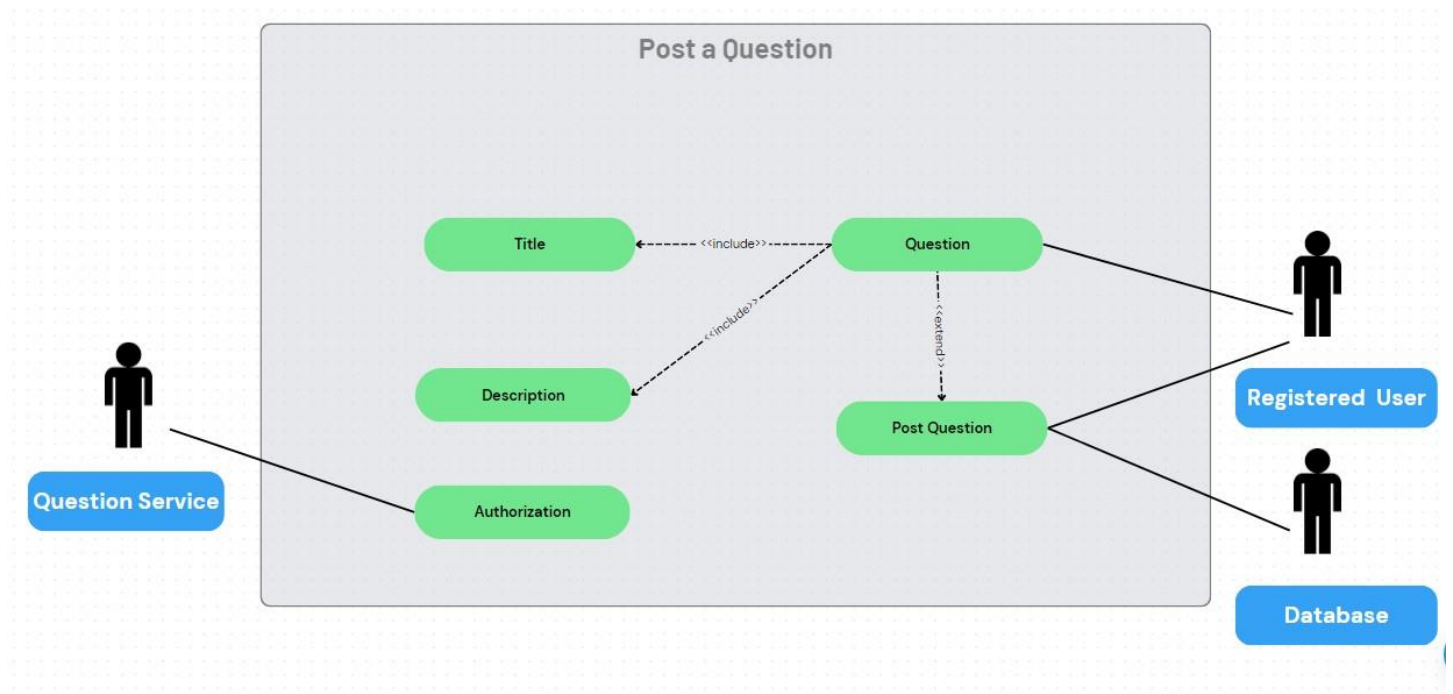
- Manage User Profile:

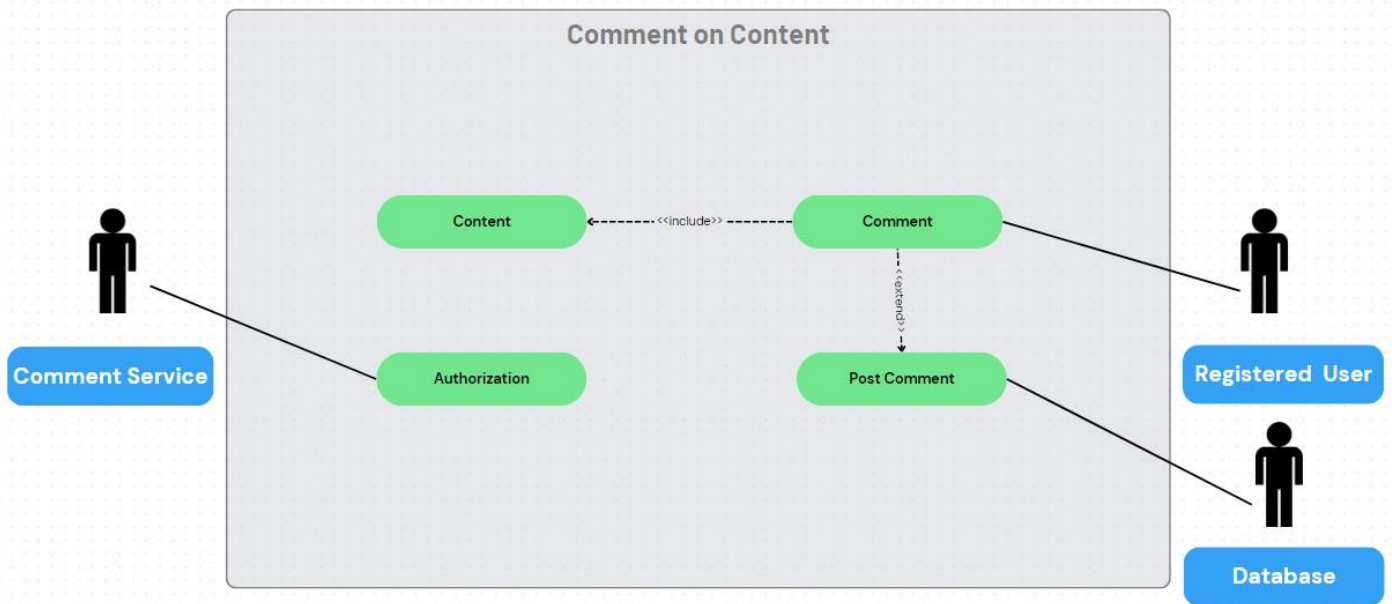
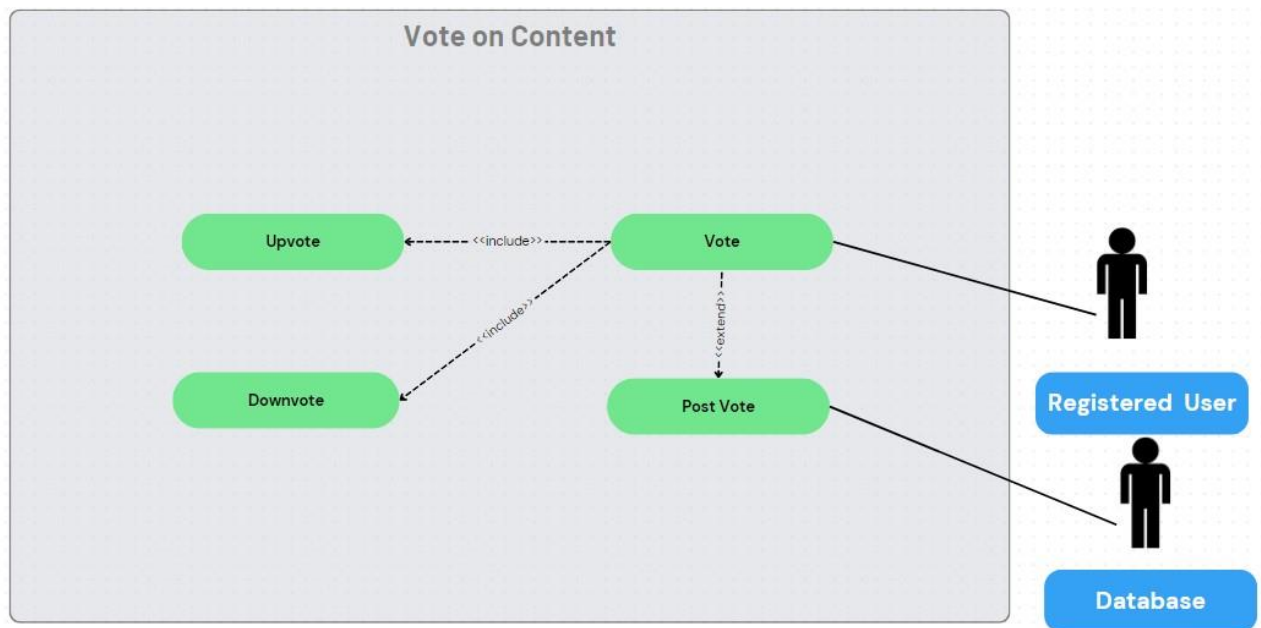
Actor: Registered User

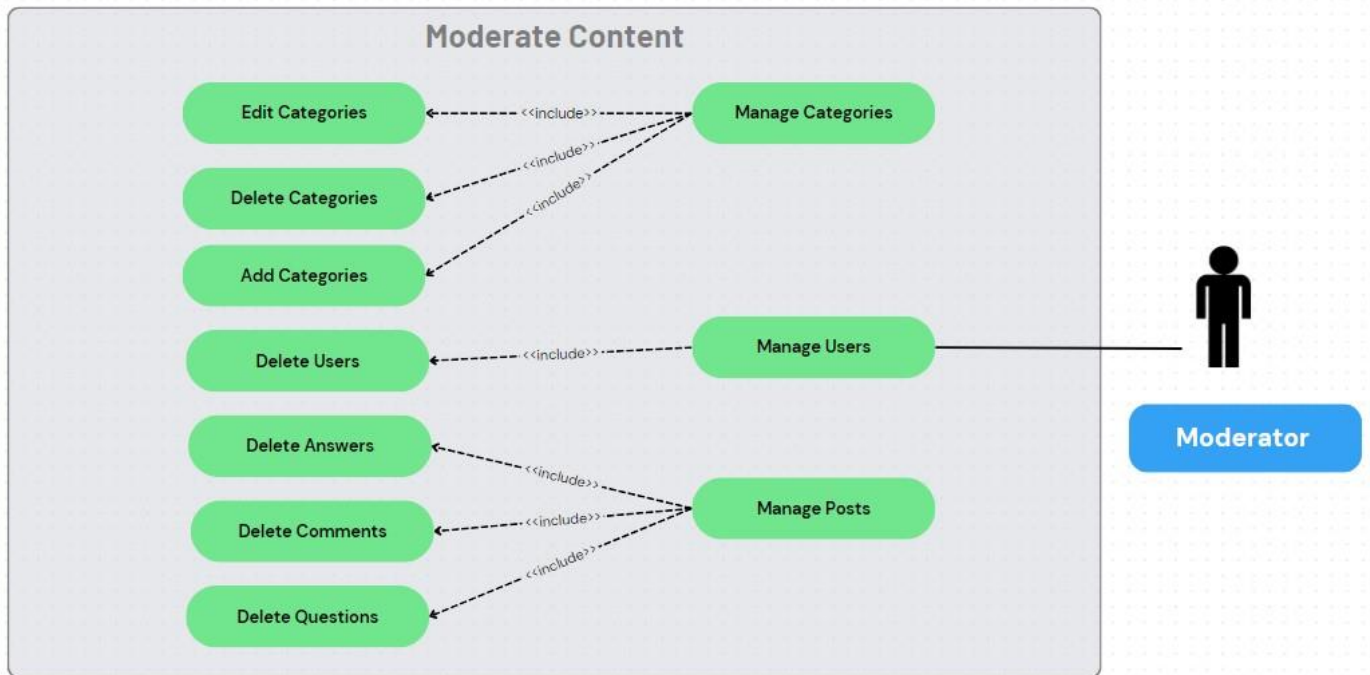
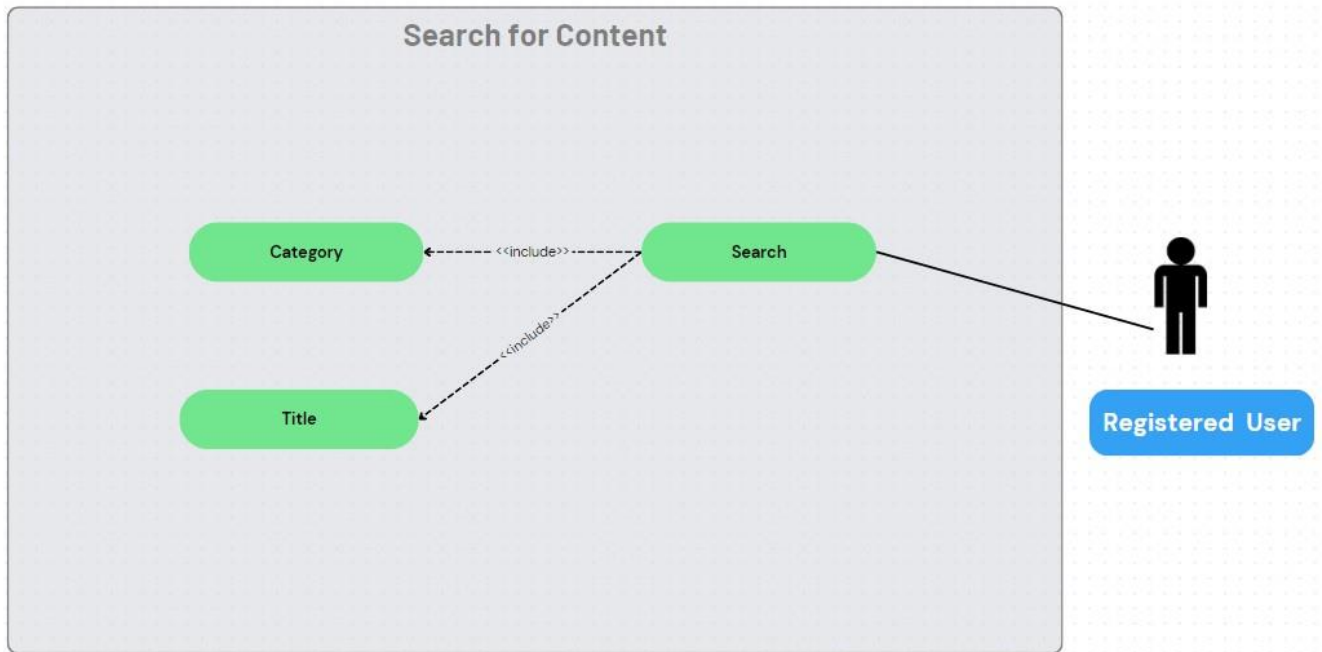
Description: A user updates their profile information and views their activity history.

## 2.2.3 Use Case Diagrams

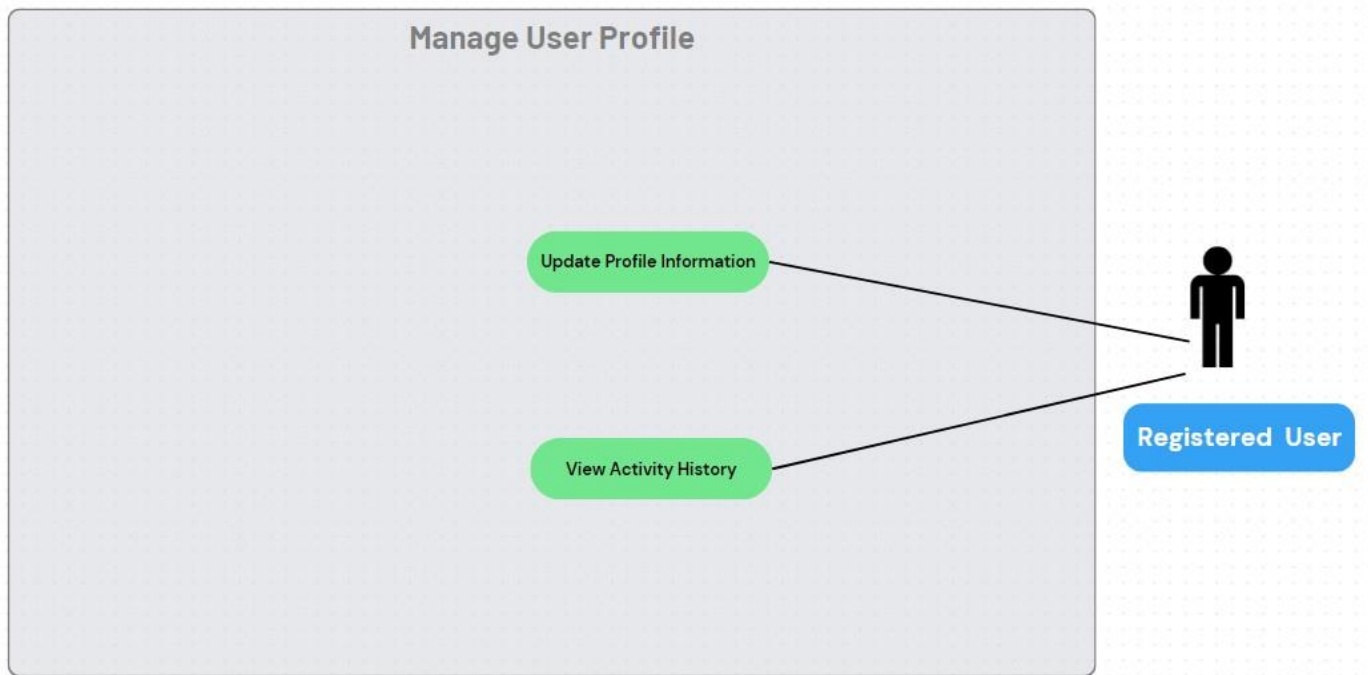












## 2.3 Classes

### 2.3.1 <List of Classes >

- **User**
- **Answer**
- **Category**
- **Comment**
- **Notification**
- **Question**

#### Class Definitions:

```
public class User extends IdentityUser
{
    public int Votes { get; set; }
```



```
}
```

```
public class Answer
```

```
{
```

```
    public int Id { get; set; }
```

```
    public int? Votes { get; set; } = 0;
```

```
    public DateTime? CreatedIn { get; set; }
```

```
    public string? Image { get; set; }
```

```
    public string? Description { get; set; }
```

```
    public int? QuestionId { get; set; }
```

```
    public string? UserId { get; set; }
```

```
    [ForeignKey("QuestionId")]
```

```
    public virtual Question? Question { get; set; }
```

```
    public virtual ICollection<Comment> Comments { get; set; } = new  
    HashSet<Comment>();
```

```
    [ForeignKey("UserId")]
```

```
    public virtual User user { get; set; }
```

```
}
```

```
public class Category
```

```
{
```

```
    public int Id { get; set; }
```

```
    public string? Name { get; set; }
```

```
    public virtual ICollection<Question> Questions { get; set; } = new  
    HashSet<Question>();
```

```
}
```

```
public class Comment
```

```
{
```

```
    public int Id { get; set; }  
    public DateTime CreatedIn { get; set; }  
    public string Content { get; set; }  
    public int AnswerId { get; set; }  
    public int UserId { get; set; }  
    public Answer Answer { get; set; }  
    public User User { get; set; }  
}
```

```
public class Notification  
{  
    public int Id { get; set; }  
    public string Message { get; set; }  
    public bool IsRead { get; set; }  
    public int UserId { get; set; }  
    public User Responder { get; set; }  
    public DateTime CreatedIn { get; set; }  
}
```

```
public class Question  
{  
    public int Id { get; set; }  
    public string Description { get; set; }  
    public string Title { get; set; }  
    public string Image { get; set; }  
    public DateTime CreatedIn { get; set; }
```

```
public int CategoryId { get; set; }  
public int UserId { get; set; }  
public Category Category { get; set; }  
public List<Answer> Answers { get; set; }  
public User User { get; set; }  
}
```

## 2.4 Non-Functional Requirements

This section details the non-functional requirements that ensure the system's performance, security, and reliability.

### 2.4.1 Performance Requirements(Availability, Security.....etc.)

- **Scalability:**

The system must handle a large number of concurrent users and transactions without performance degradation.

- **Availability:**

The platform should be available 99.9% of the time, with minimal downtime for maintenance.

- **Security:**

User data must be protected using encryption (e.g., HTTPS, SSL). Implement robust authentication and authorization mechanisms to prevent unauthorized access. Regular security audits and vulnerability assessments.

- **Performance:**

Page load times should be under 2 seconds under normal usage conditions by depending on in-memory caching database (Redis)

- **Usability:**

The platform should have an intuitive and user-friendly interface. Responsive design to ensure usability across different devices (desktop, tablet, mobile).

### 3. Objectives/List of Services(measurable)

The primary objectives of this question-answering platform are to provide a robust, user-friendly environment where users can efficiently seek and share knowledge. The objectives are measurable and time-bound to ensure progress can be tracked and evaluated effectively.

#### 3.1.1 Improve User Engagement

- Objective: Increase the average session duration by 20% within the first six months.
- Measurement: Track user session lengths through analytics tools and compare with baseline data.

#### 3.1.2 Enhance Content Quality

- Objective: Achieve a 90% approval rating on answers (upvotes vs. downvotes) within the first year.
- Measurement: Monitor the ratio of upvotes to downvotes on answers.

#### 3.1.3 Increase User Base

- Objective: Register 10,000 new users within the first year.
- Measurement: Track the number of new registrations monthly.

#### 3.1.4 Improve Response Time

- Objective: Reduce the average time to receive an answer by 30% within six months.

- Measurement: Measure the time between question posting and first answer received.

### 3.1.5 Ensure System Reliability

- Objective: Maintain 99.9% system uptime in the first year.
- Measurement: Use monitoring tools to track uptime and downtime.

## 3.2 LIST OF SERVICES (MEASURABLE)

This section outlines the specific services provided by the platform, each with measurable targets to ensure their effectiveness and contribution to the overall objectives.

### 3.2.1 User Registration and Profile Management

- Service: Enable users to register and manage their profiles.
- Measurement:
  - Target: Register 1,000 new users per month.
  - Metric: Number of new registrations tracked monthly.
- Action: Enhance the registration process, ensure security, and offer profile customization features.

### 3.2.2 Question and Answer Posting

- Service: Allow users to post questions and answers.
- Measurement:
  - Target: Achieve an average of 500 new questions and 1,500 new answers per month.
  - Metric: Number of questions and answers posted tracked monthly.

- Action: Promote user engagement through incentives and ensure a user-friendly posting process.

### 3.2.3 Voting and Reputation System

- Service: Provide a voting system to rate questions and answers, and a reputation system to reward users.
- Measurement:
  - Target: 80% of users to participate in voting within the first year.
  - Metric: Number of votes cast tracked monthly.
- Action: Educate users on the importance of voting and integrate visual indicators of reputation to encourage participation.

### 3.2.4 Search and Filtering

- Service: Enable advanced search and filtering options for finding relevant content.
- Measurement:
  - Target: Achieve a search success rate of 90% (users find what they are looking for).
  - Metric: User feedback and search success metrics tracked quarterly.
- Action: Optimize search algorithms and improve the tagging system.

### 3.2.5 Notification System

- Service: Provide notifications for user activities and relevant updates.
- Measurement:

- Target: 95% of active users to receive and engage with notifications.
- Metric: Notification engagement rates tracked monthly.
- Action: Implement personalized notifications and ensure timely delivery.

### 3.2.6 Moderation and Content Quality Control

- Service: Implement moderation tools to maintain content quality.
- Measurement:
  - Target: 98% of flagged content to be reviewed within 24 hours.
  - Metric: Time taken to review flagged content tracked daily.
- Action: Recruit and train moderators, and implement automated moderation tools.

### 3.2.7 Help and Support

- Service: Offer comprehensive help and support resources.
- Measurement:
  - Target: Resolve 90% of user support queries within 48 hours.
  - Metric: Response and resolution times tracked weekly.
- Action: Develop detailed help documentation to guide users.



## 4. Design Overview

### 4.1 System Architecture

The system architecture for the question-answering platform is based on Onion Architecture coupled with CQRS (Command Query Responsibility Segregation). This combination ensures a clean separation of concerns, promotes maintainability, and enhances scalability.

#### **Union Architecture Layers:**

##### **1. Domain Layer:**

- a) Entities: Represent the fundamental business objects.
- b) JWT settings :Represent the necessary properties of JWT configuration.

##### **2. Infrastructure Layer:**

- Repositories: Interfaces for data access operations, implemented in this layer.
- DB context : the configuration of entity framework and the models by fluent API.

##### **3. Service Layer:**

Services: Interfaces for the business services operations, implemented in this layer (e.g., real time, caching).

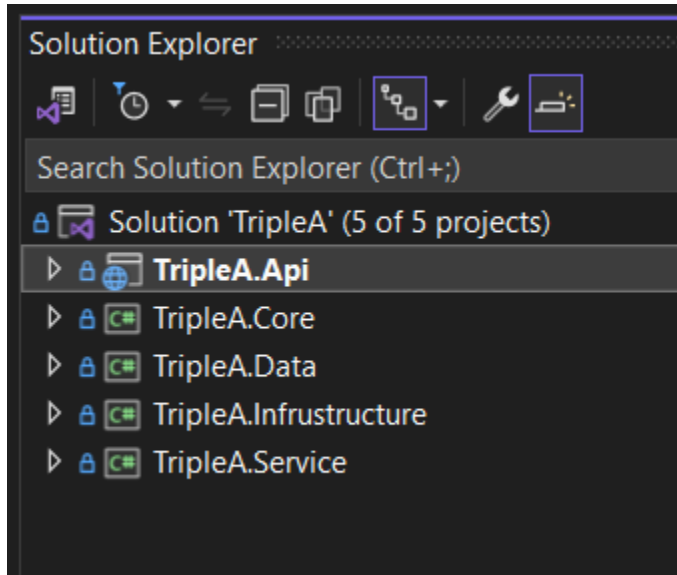
##### **4. Core Layer:**

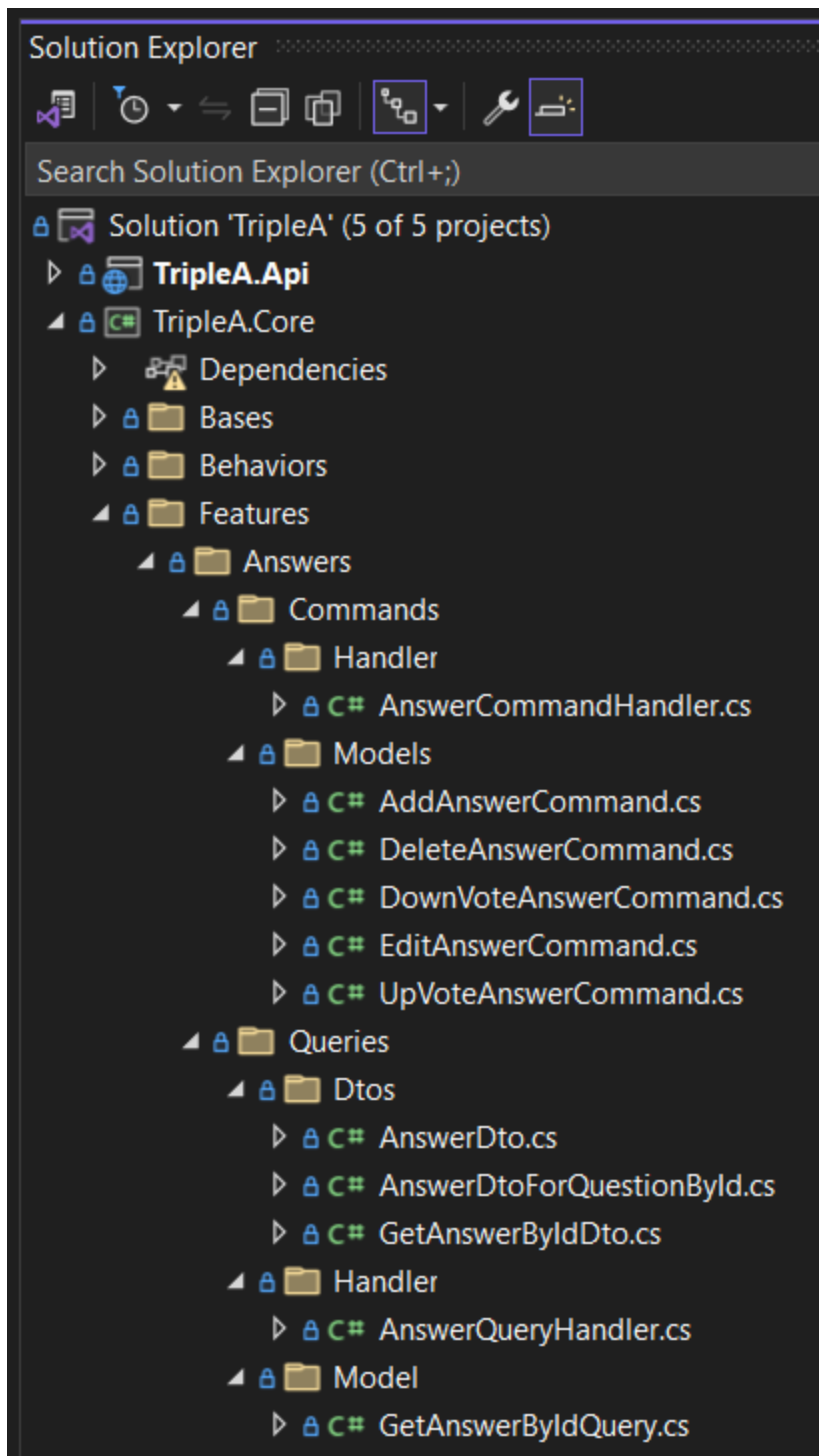
- a) Commands: Represent intentions to change the state of the system.

- b) Queries: Represent requests to retrieve data without altering state.
- c) DTOs (Data Transfer Objects): Used for transferring data between processes.

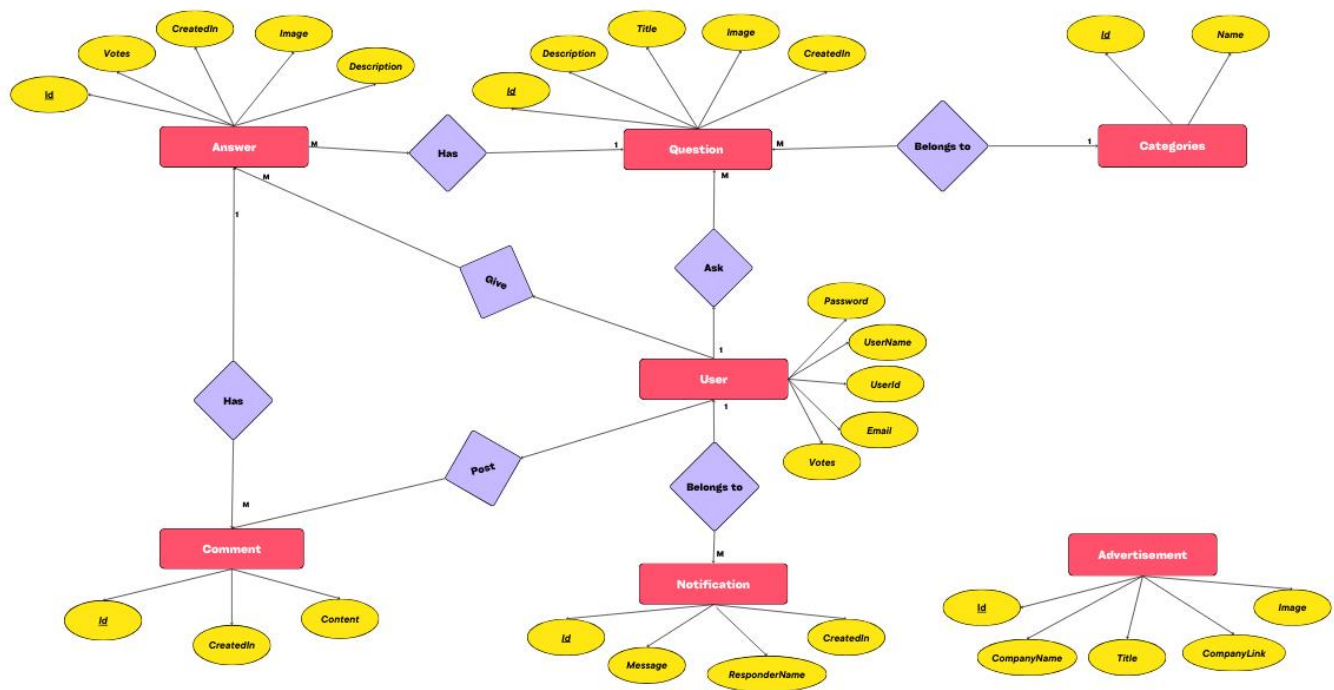
## 5. Presentation Layer:

APIs: RESTful endpoints for interacting with the application.





## 4.2 Data Design(Entity Relationship Diagram)



### Implementation of Database:

Creating a database for an Ask and Answer typically involves defining tables to store information about the Question, Answer, Category, Comment, and Notification. Once we defined the table schema, we populated the tables with data. We used the Bulk Insert command. This command allows for the efficient insertion of large amounts of data into tables from a data file. The data file can be CSV in our system.

0 references

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);

    modelBuilder.Entity<Comment>()
        .HasOne(c => c.Answer)
        .WithMany(p => p.Comments)
        .HasForeignKey(c => c.AnswerId)
        .OnDelete(DeleteBehavior.Cascade);

    modelBuilder.Entity<Answer>()
        .HasOne(c => c.Question)
        .WithMany(p => p.Answers)
        .HasForeignKey(c => c.QuestionId)
        .OnDelete(DeleteBehavior.Cascade);
}
```

0 references

```
public DbSet<Category> Categories { get; set; }
```

0 references

```
public DbSet<Question> Questions { get; set; }
```

0 references

```
public DbSet<Answer> Answers { get; set; }
```

0 references

```
public DbSet<Comment> Comments { get; set; }
```

0 references

```
public DbSet<Notification> Notifications { get; set; }
```

## 5. Implementation

### 5.1 Tools & Technologies



### 5.2 Hardware Requirements

Since the project is a website, the hardware requirements for end-users are minimal. Users need a device capable of browsing the internet and accessing modern web applications. Below are the recommended and minimum hardware specifications:

#### 5.2.1 End-User Hardware Requirements

**Recommended Specifications:**

**Device:** Modern PC, laptop, tablet, or smartphone

**Processor:** Dual-core CPU or better

RAM: 4 GB or more

Storage: 100 MB of free storage space for caching

Display: Screen resolution of 1280x720 or higher

Internet Connection: Broadband connection with at least 5 Mbps download speed

Minimum Specifications:

Device: Any device capable of running a modern web browser

Processor: Single-core CPU

RAM: 2 GB

Storage: 50 MB of free storage space for caching

Display: Screen resolution of 1024x768

Internet Connection: Stable internet connection with at least 1 Mbps download speed

## 5.3 Steps of Installation

### 5.3.1 Setting Up the Development Environment (.Net API)

#### 1. Install Development Tools:

- Install Visual Studio or Visual Studio Code.
- Install Git for version control.
- Install Docker for containerization (optional but recommended for Redis).

#### 2. Clone the Repository:

- Use Git to clone the project repository:

`git clone https://github.com/AhmedReda103/TripleA.git`

### 3. Install Dependencies:

- Navigate to the project directory and restore necessary dependencies:

`cd question-answer-platform`

`dotnet restore`

### 4. Set Up Environment Variables:

- Create a `appsettings.Development.json` file in the project root and configure necessary environment variables (e.g., database connection strings, API keys).

### 5. Configuring the Database

#### 1. Choose a Database:

- Select a database management system (SQL Server).

#### 2. Install the Database:

- Follow the database installation guide for your operating system or use a managed database service.

#### 3. Run Migrations:

- Apply database migrations to set up the schema:  
`dotnet ef database update`

### 6. Deploying the Application

#### 1. Build the Application:

- Compile and build the application:  
`dotnet build`



## 7. Start the Server:

- Start the application server:

```
dotnet run
```

## 5.3.2 Setting Up the Angular Frontend

### 1. Install Node.js and npm:

- Download and install Node.js from the official website.
- Verify the installation by running:

```
node -v
```

```
npm -v
```

### 2. Install Angular CLI:

- Install Angular CLI globally using npm:

```
npm install -g @angular/cli
```

### 3. Clone the Repository:

- Use Git to clone the project repository (if not already done):

```
git clone https://github.com/ZeyadMohamed1805/TripleA.git
```

### 4. Navigate to the Angular Project Directory:

- Change to the Angular project directory:

```
cd question-answer-platform/frontend
```

### 5. Install Dependencies:

- Install the required npm packages:

```
npm install
```

### 6. Running the Angular Application

## Start the Development Server:

- Run the Angular development server:  
`ng serve`
- By default, the Angular app will be served at <http://localhost:4200>.

## 6. User/Customer Testing

### 6.1 Users/Customers Feedback

- User Interface (UI) and User Experience (UX):
  - Users appreciated the clean and intuitive design but suggested enhancements for navigation and accessibility.
- Community Engagement:
  - Users valued the community aspect but recommended features to enhance interaction, such as more comprehensive user profiles.

### 6.2 Improvements done

Based on the feedback received from users and customers, several improvements were made to the platform to enhance its usability, performance, and overall user satisfaction.

#### 6.2.1 User Interface (UI) and User Experience (UX) Improvements

- Navigation Enhancements:
  - o Redesigned the navigation menu to make it more intuitive and easier to access frequently used features.
  - o Improved the visibility and accessibility of important features through better placement and highlighting.

#### 6.2.2 Community Engagement Features

- User Profiles and Social Features:

- Enhanced user profiles to include more detailed information and added social features such as private messaging and user activity feeds.

### 6.2.3 Performance and Speed Improvements

- Caching Implementation:
  - Implemented caching mechanisms to store frequently accessed data and reduce the load on the server, resulting in faster response times.

By implementing these improvements, the platform has become more user-friendly, efficient, and engaging, meeting the needs and expectations of its users and customers more effectively. Continuous feedback and iteration will ensure the platform remains a valuable resource for its community.

## 7. Recommendations(Future Work)

Enhance Search Capabilities Improving the search functionality can significantly enhance user experience.

Consider implementing the following:

- Advanced Search Filters: Allow users to filter search results by tags, date range, number of votes, etc.
- Full-Text Search: Use a full-text search engine like Elasticsearch to improve search performance and accuracy.
- Premium Account: Allows you to hide ads and access the high popular answers.

## 8. User Guide(See Attachment)

### 8.1 Quick Start Guide

Welcome to the Quick Start Guide for the Question-Answering Platform. This guide will help you get started with using the platform quickly and efficiently.

#### 8.1.1 Registration

1. Access the Registration Page:

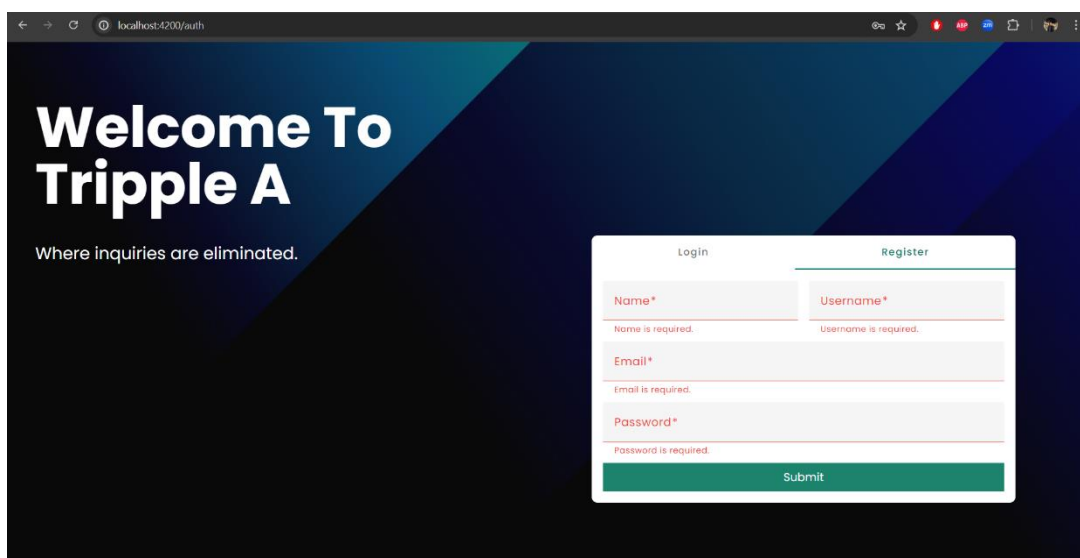
- Navigate to the registration page by accessing the url of the site and clicking Registration button.

2. Fill in Registration Details:

- Enter your username, email address, and password.

3. Complete Registration:

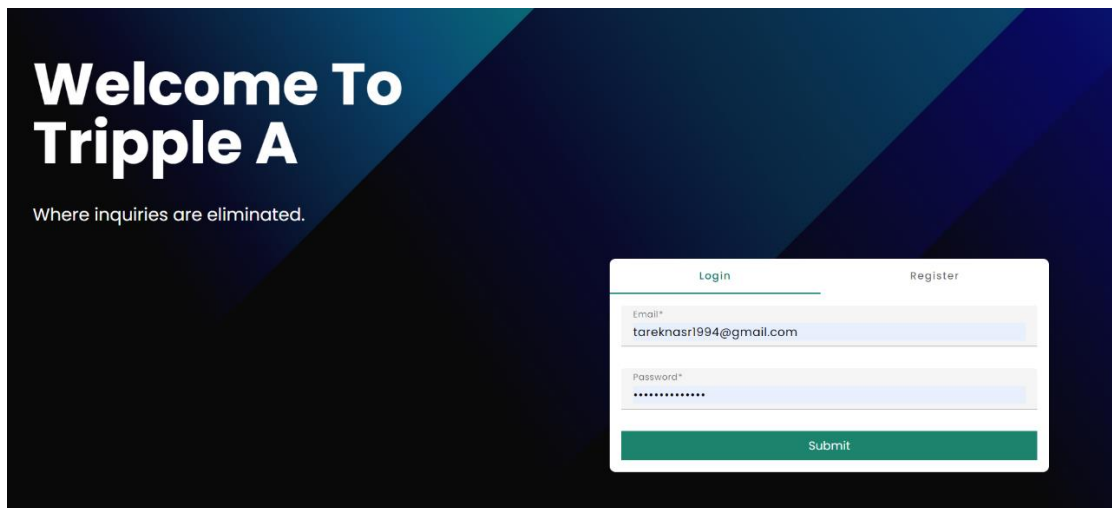
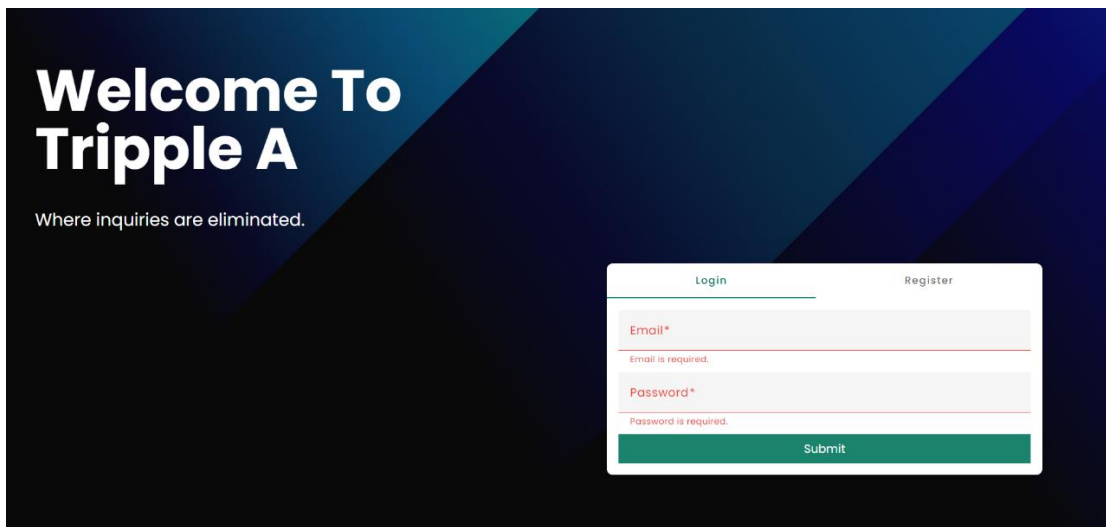
- Click the "Register" button to create your account.



The screenshot shows a web browser window at localhost:4200/auth. The page features a dark blue background with a diagonal split. On the left, the text 'Welcome To Trippl A' is displayed in white, with the tagline 'Where inquiries are eliminated.' below it. On the right, there is a white registration form. The form has two tabs: 'Login' and 'Register', with 'Register' being the active tab. The form contains three input fields: 'Name\*', 'Email\*', and 'Password\*'. Each field has a red border and a red error message below it: 'Name is required.', 'Email is required.', and 'Password is required.' respectively. At the bottom of the form is a green 'Submit' button.

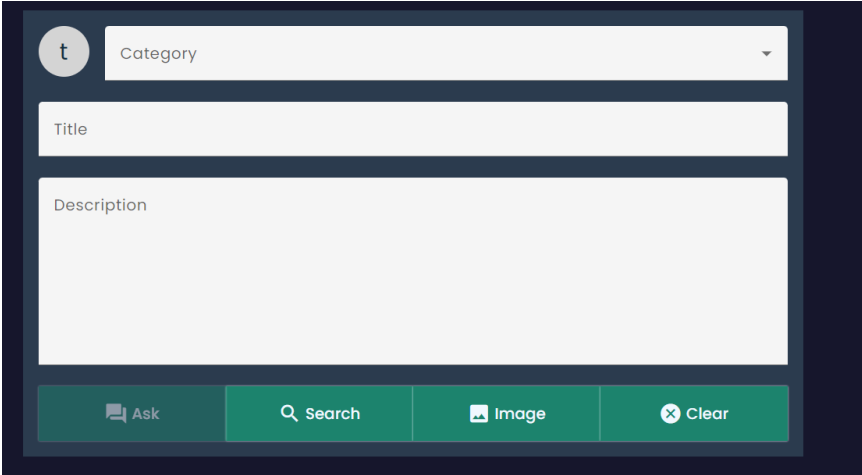
## 8.1.2 Logging In

1. Access the Login Page:
  - Click the "Login" button on the homepage.
2. Enter Login Credentials:
  - Enter your registered email address and password.
3. Complete Login:
  - Click the "Login" button to access your account dashboard.



### 8.1.3 Asking a Question

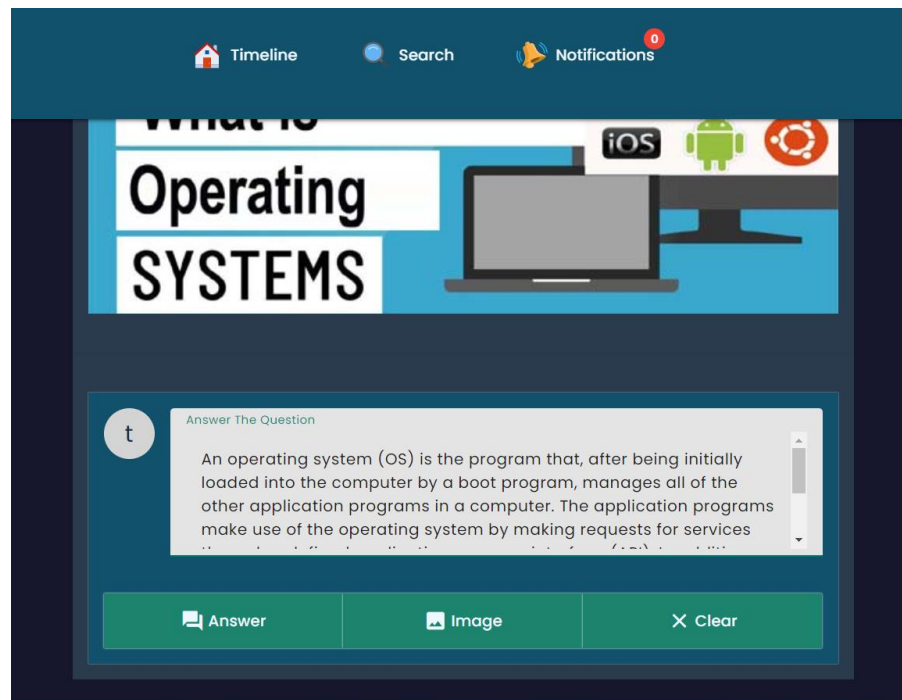
1. Navigate to Ask Question Page:
  - Click the "Home" button on the top navigation bar.
2. Enter Question Details:
  - Provide a descriptive title for your question.
  - Enter the details of your question in the text area.
  - Add relevant tags to categorize your question.
3. Submit Question:
  - Click the "ASK" button to post your question.

A screenshot of a web form for asking a question. The form is set against a dark blue background. It features a circular profile icon with the letter 't' on the left. To its right is a 'Category' dropdown menu. Below these are three input fields: 'Title', 'Description' (a larger text area), and a bottom bar with four buttons: 'Ask' (with a speech bubble icon), 'Search' (with a magnifying glass icon), 'Image' (with a picture icon), and 'Clear' (with an 'x' icon).

### 8.1.4 Answering a Question

1. Browse Questions:
  - Navigate to the "Questions" page to browse available questions.
2. Select a Question:
  - Click on a question you would like to answer.
3. Provide an Answer:
  - Enter your answer in the provided text area.
  - Click the "Submit Answer" button to post your answer.





## 8.2 Main Scenarios

This section outlines the main scenarios users may encounter while using the platform.

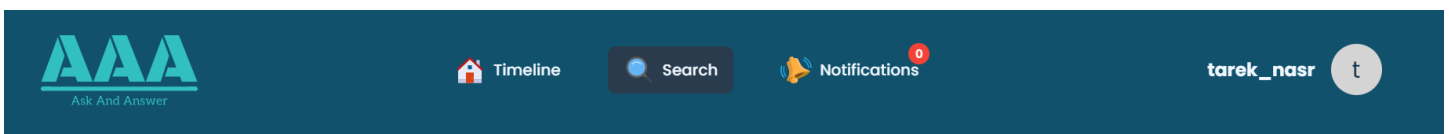
### 8.2.1 Searching for Answers

1. Access the Search Bar:
  - Use the search icon at the top of the page to access the search page.
2. Apply Filters:
  - Use filters to narrow down search results by Category or Title.
3. View Results:
  - Browse through the search results and click on a question to view detailed answers.

The image displays two variations of a search interface. The top variation features a dark blue header with 'Title' and 'Category' labels. Below 'Title' is a white text input field, and below 'Category' is a green button labeled 'Search'. The bottom variation features a similar dark blue header, but the 'Category' label is positioned above a white dropdown menu. The 'Search' button remains green and is located below the dropdown.

### 8.2.2 Accessing the notification

Click the ring icon on the top of the page to access all notifications are sent to you



### 8.2.3 Voting on Answers

Click the upvote or downvote arrow next to an answer to indicate its quality.

t

**tarek\_nasr**  
Jun 30, 2024

0

An operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer. The application programs make use of the operating system by making requests for services through a defined application program interface (API).

Upvote

Comment

Downvote

## 8.3 Troubleshooting

This section provides solutions to common issues users might face.

### 8.3.1 Posting Issues

- Question Not Posting:
  - Ensure all required fields are filled in and adhere to the community guidelines.
- Answer Not Submitting:
  - Check your internet connection and try reloading the page.
  - Ensure all fields are correctly filled out and try again. If the issue persists, contact support.

### 8.3.2 Search Issues

- No Search Results:
  - Broaden your search criteria and ensure there are no typos in your keywords.