SOFTWARE ENGINEERING

FINAL PROJECT REPORT

INDIAN INSTITUTE OF TECHNOLOGY, MADRAS

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE COURSE:

BSCSS3001: Software Engineering

By:

*Aditya Raj (**21f1003293**)*
*Himanshu Singh (**21f1003237**)*
*Sachin Sinha (**21f1006475**)*

Identifying the various types of Users

Primary Users:

- Students
- Support Teams

Secondary Users:

- IIT Madras B.S Degree Administration

Tertiary Users:

- A deep learning model can be trained on this dataset to answer questions using NLP.
- Data Analytics Team

User Stories for the identified users:

Primary User : Student

- As a student , I want to be able to create new ticket so that I can solve my problems.

- As a student, I want to be able to search for similar tickets before creating a new one so that my answers so that I can know the answer of doubt if it is already answered.
- As a student, I want to be able to like (+1) an existing ticket if it's similar to the concern or query I have, so that the support team can prioritize the frequently asked concerns.
- As a student, I want to be able to see the current status of my ticket and solutions given by the support team so that I can get updates on the progress of my query.
- As a student, I want to be able to see the history of my previous tickets so that I can keep track of my old queries.
- As a student , I should be able to mark my queries as operational or course related so that it would be helpful for the support team.

Primary User: Support team

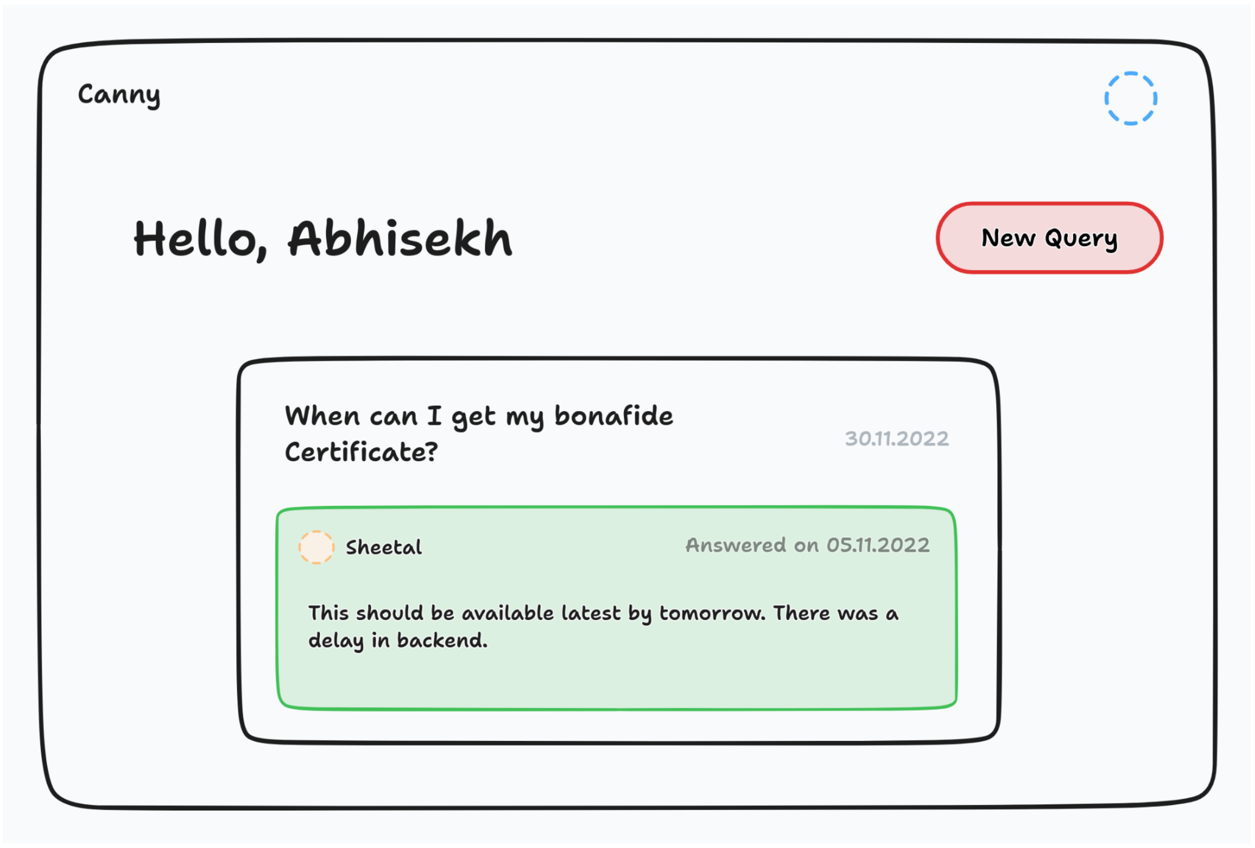
- As a support team member, I want to be able to see a sorted list of all the support tickets filtered by upvotes and last update so that I can respond to them in a timely manner.

- As a support team member, I want to be able to change the status of a support ticket so that the ticket is updated.
- As a support team member, I want to redirect the queries to the appropriate team member so that they can resolve the queries efficiently.

Secondary User: Administration

- As an administrator, I want to be able to see the history of support tickets arranged by upvotes and time so that I can understand the queries of the students.
- As an administrator, I want to be able to assign different team members to different support teams so that they will be able to see only the tickets assigned to them.
- As an administrator, I want to manage the FAQ section, so that students have access to up-to-date information about common concerns and queries.
- As an administrator, I want to be able to enroll a support team to the portal directly using their email IDs so that they can use the portal to resolve queries.

MILESTONE 2


Canny

Hello, Abhisekh

New Query

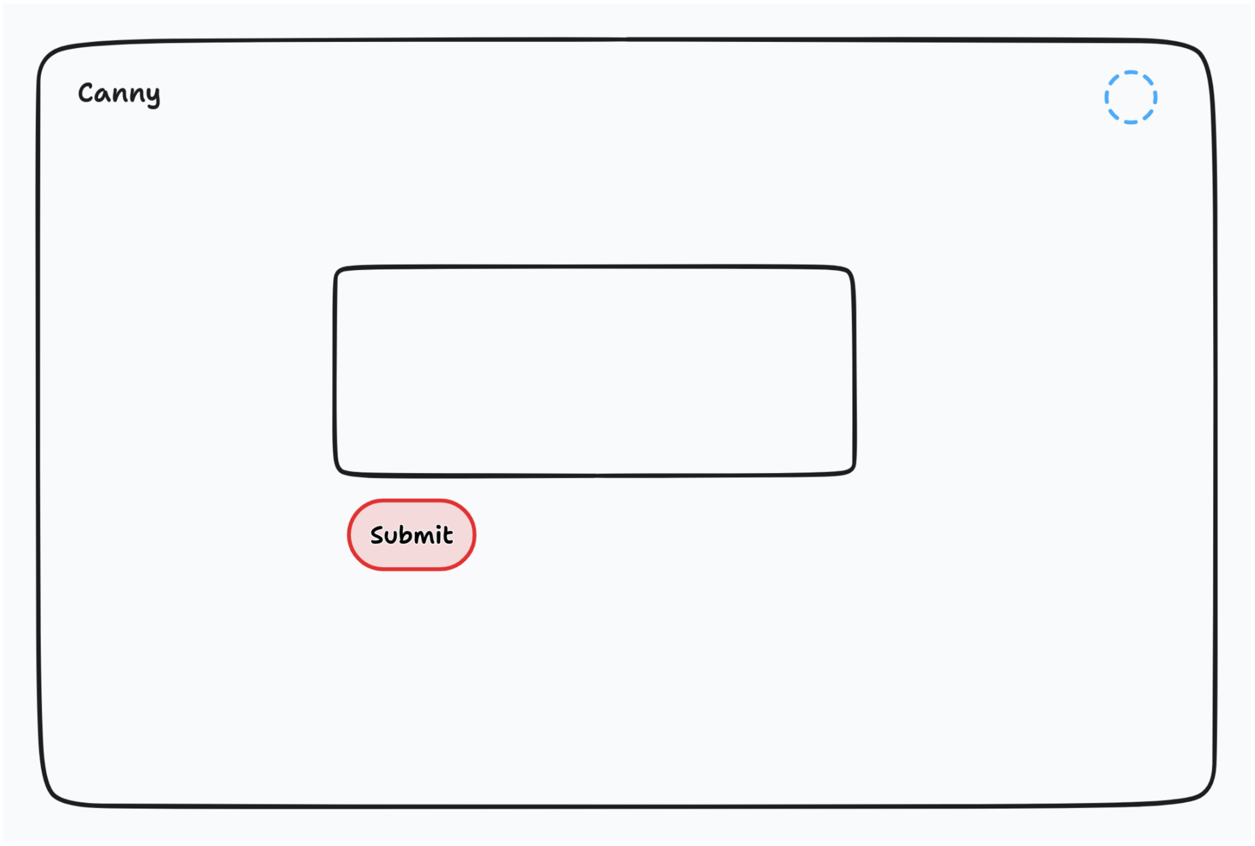
When can I get my bonafide Certificate?

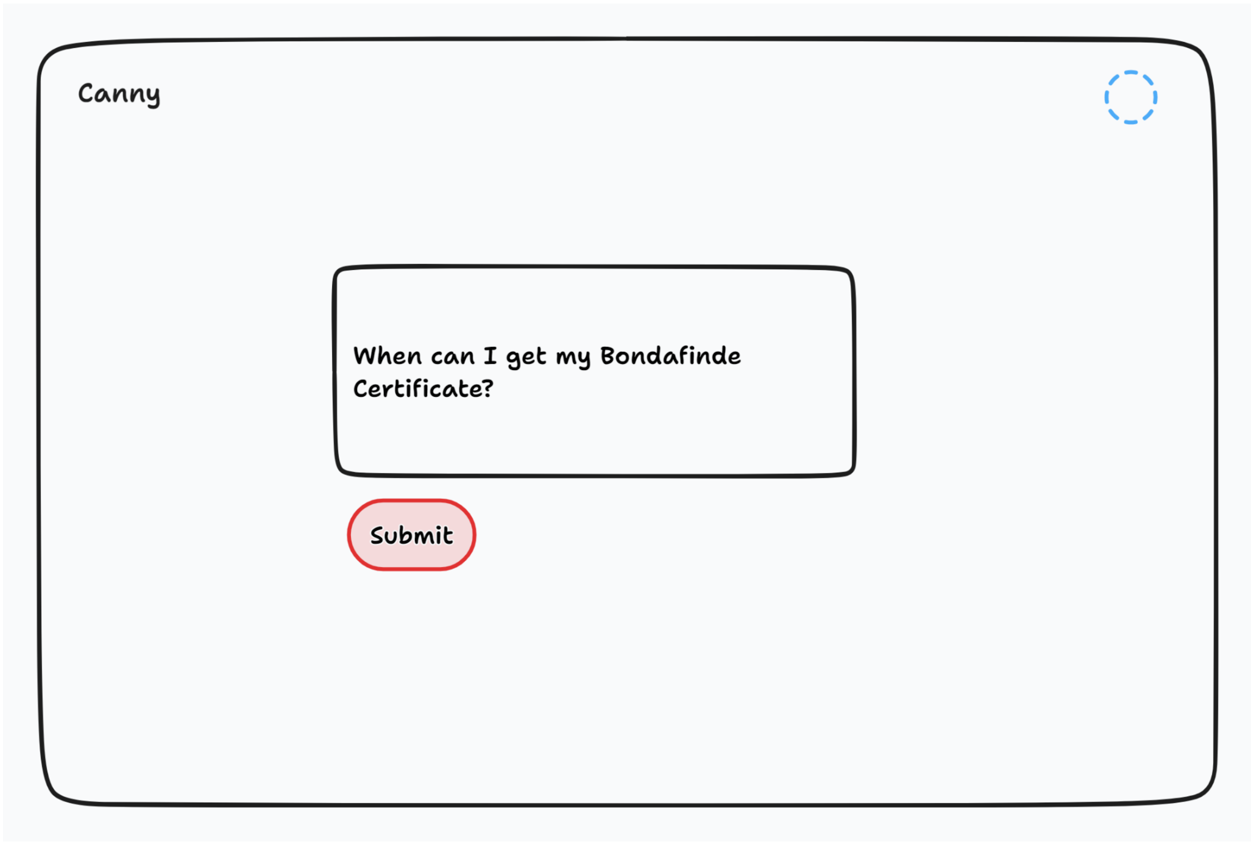
30.11.2022

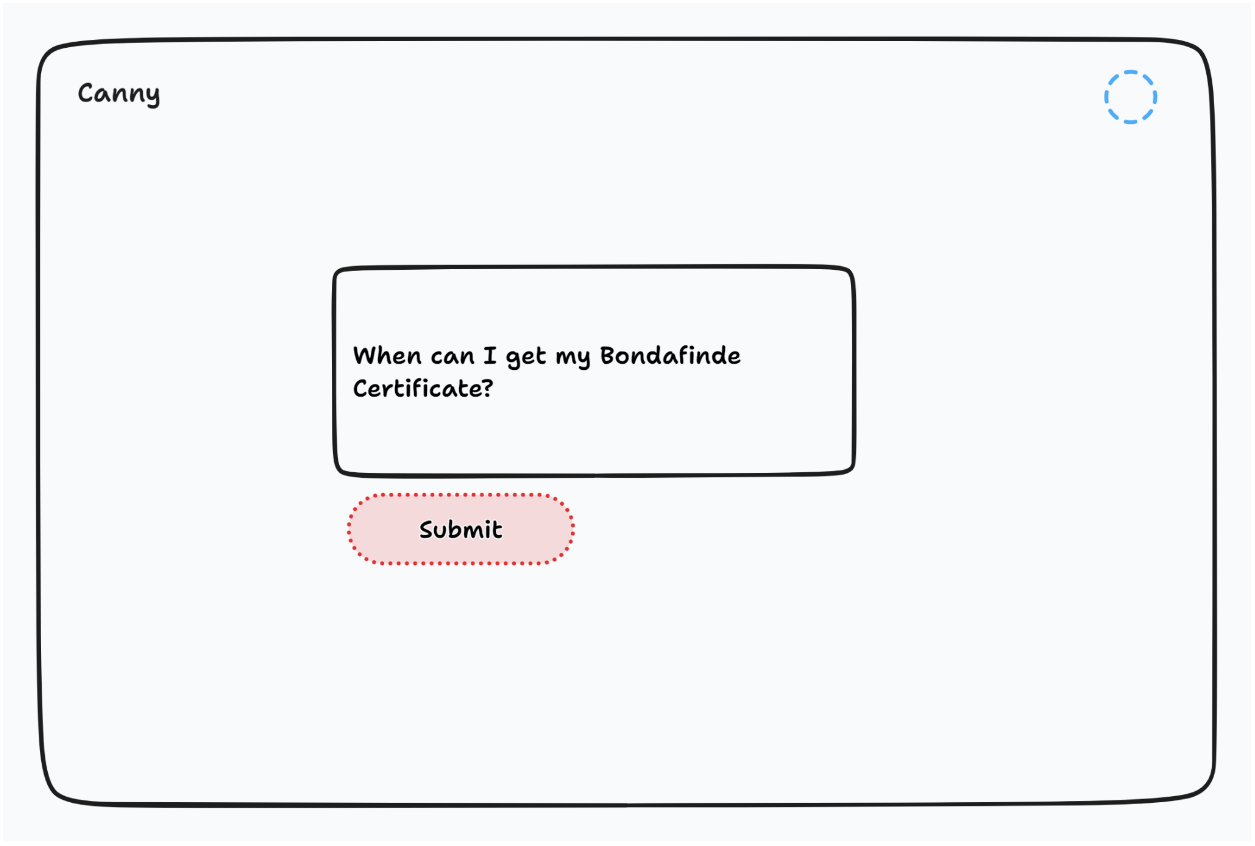
Sheetal

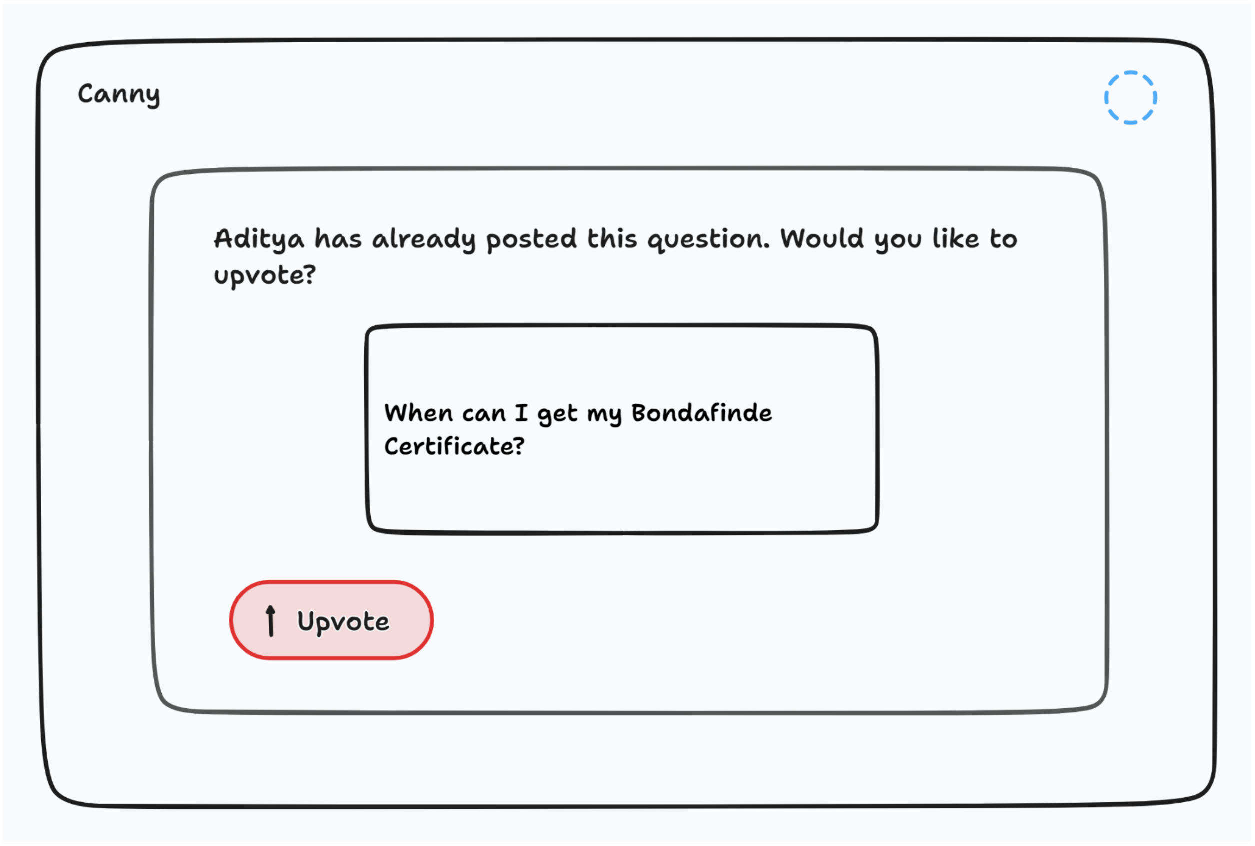
Answered on 05.11.2022

This should be available latest by tomorrow. There was a delay in backend.


Canny
Submit

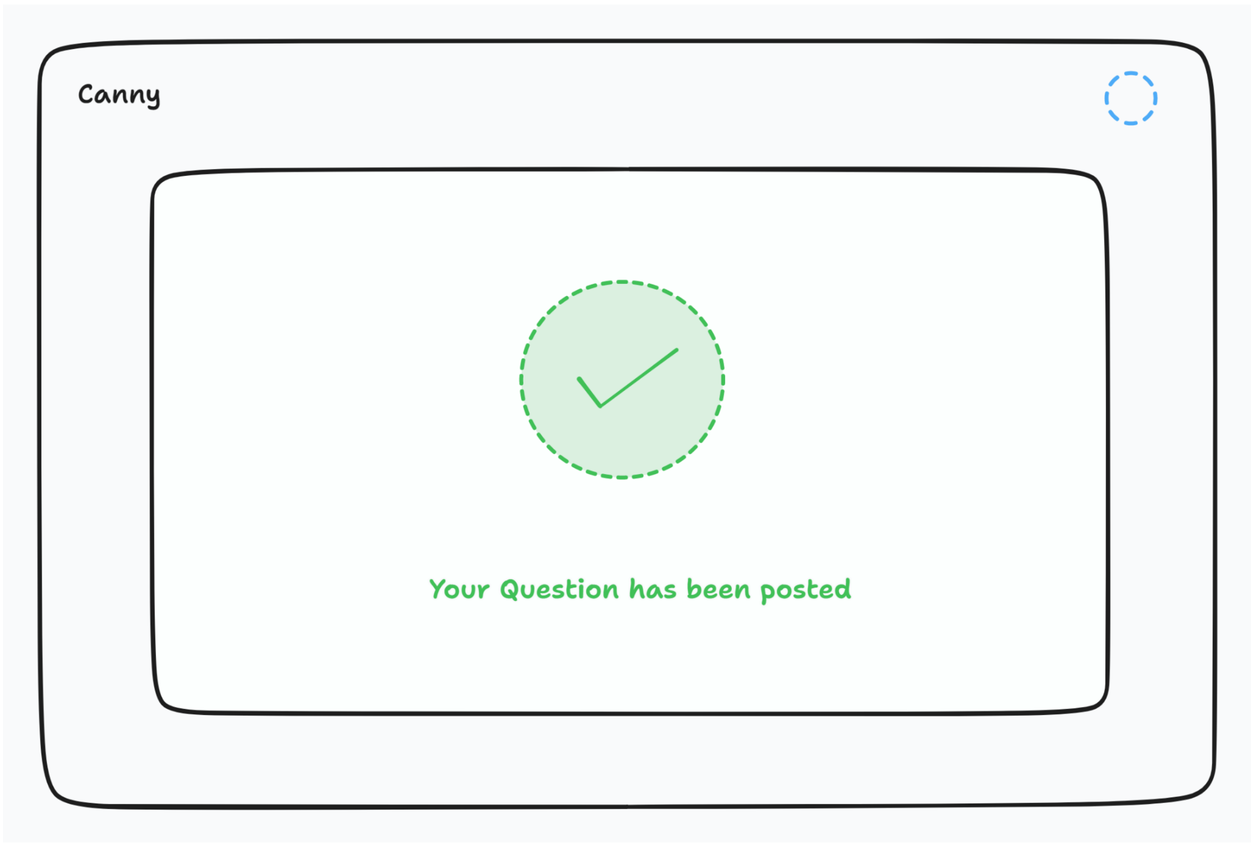

Canny
When can I get my Bondafinde Certificate?
Submit


Canny
When can I get my Bondafinde Certificate?
Submit



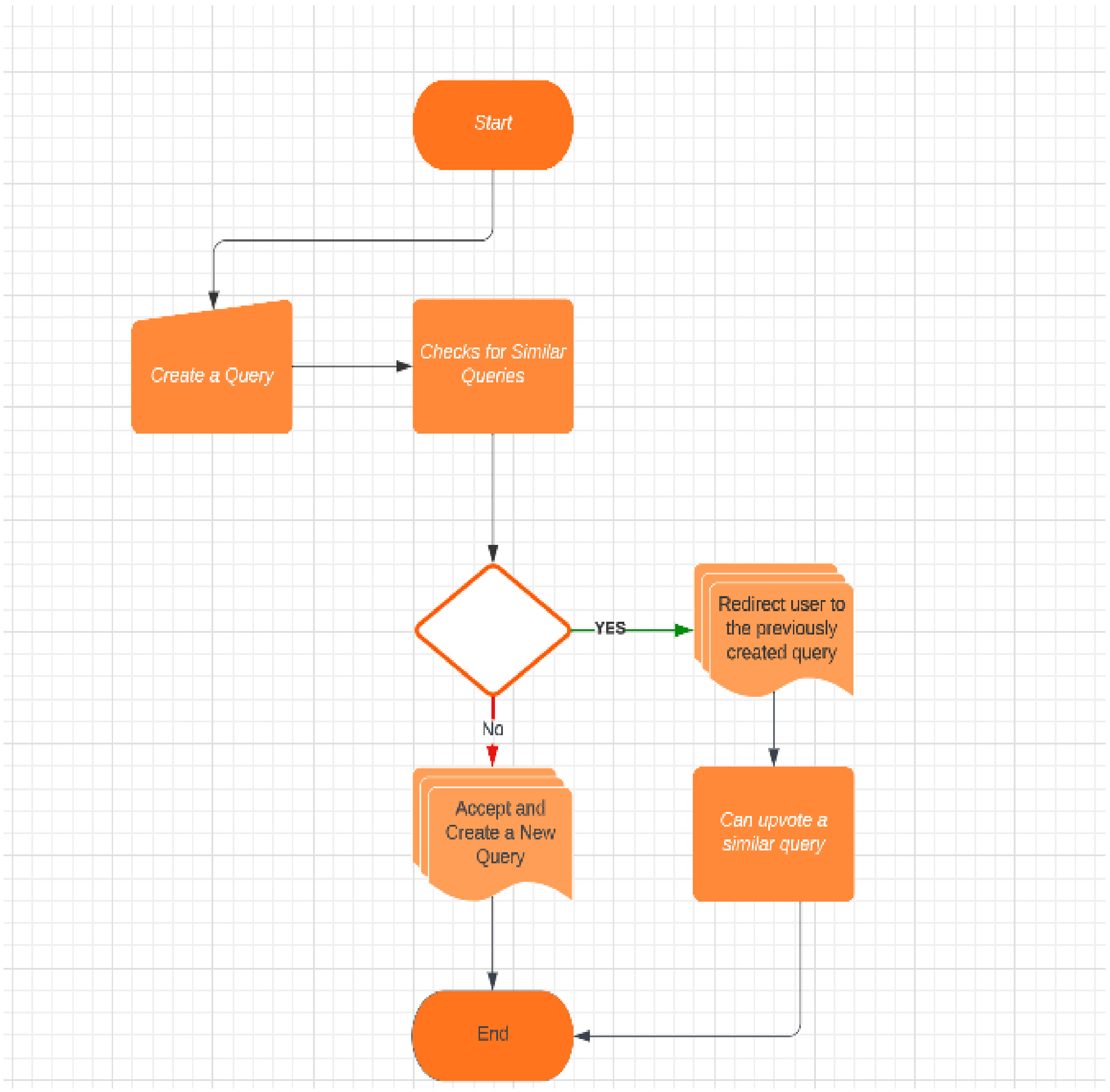
Aditya has already posted this question. Would you like to upvote?

When can I get my Bondafinde Certificate?

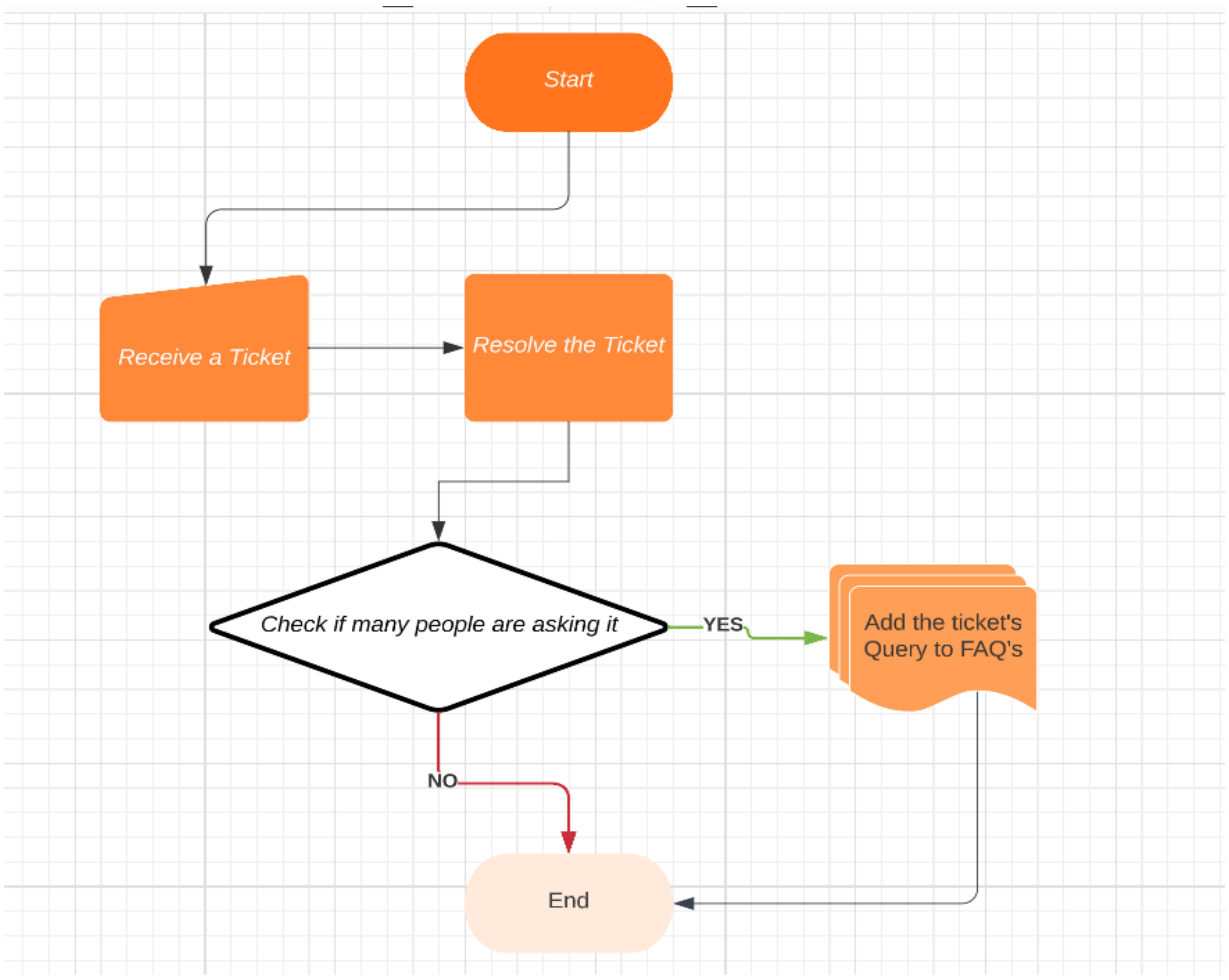

Canny
Your Question has been posted

MILESTONE 3

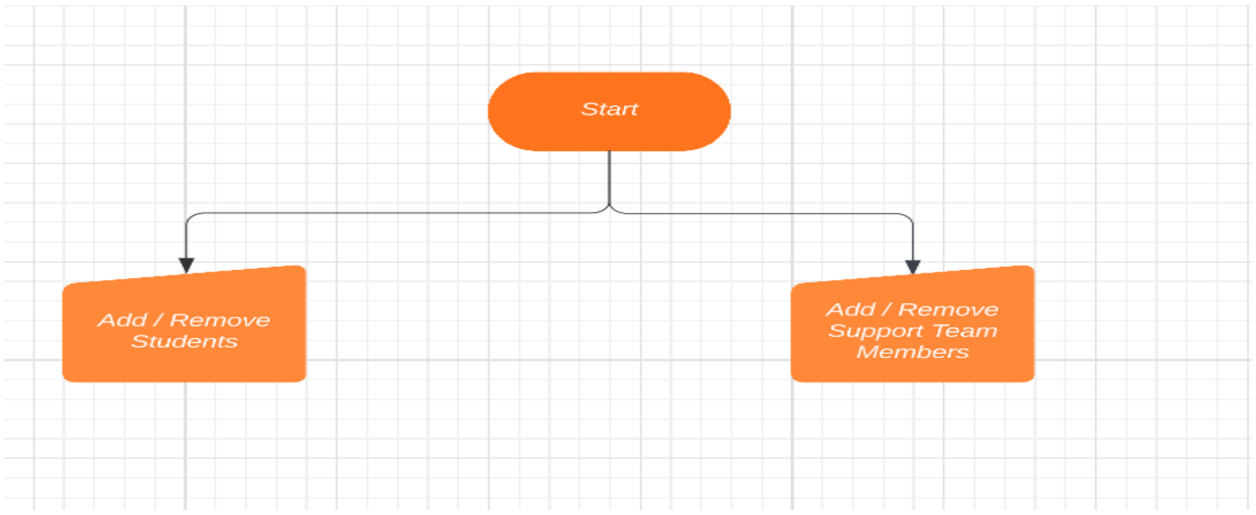
User Flow Diagrams


Start
Create a Query
Checks for Similar Queries
YES
Redirect user to the previously created query
No
Accept and Create a New Query
Can upvote a similar query
End

Student User Flow

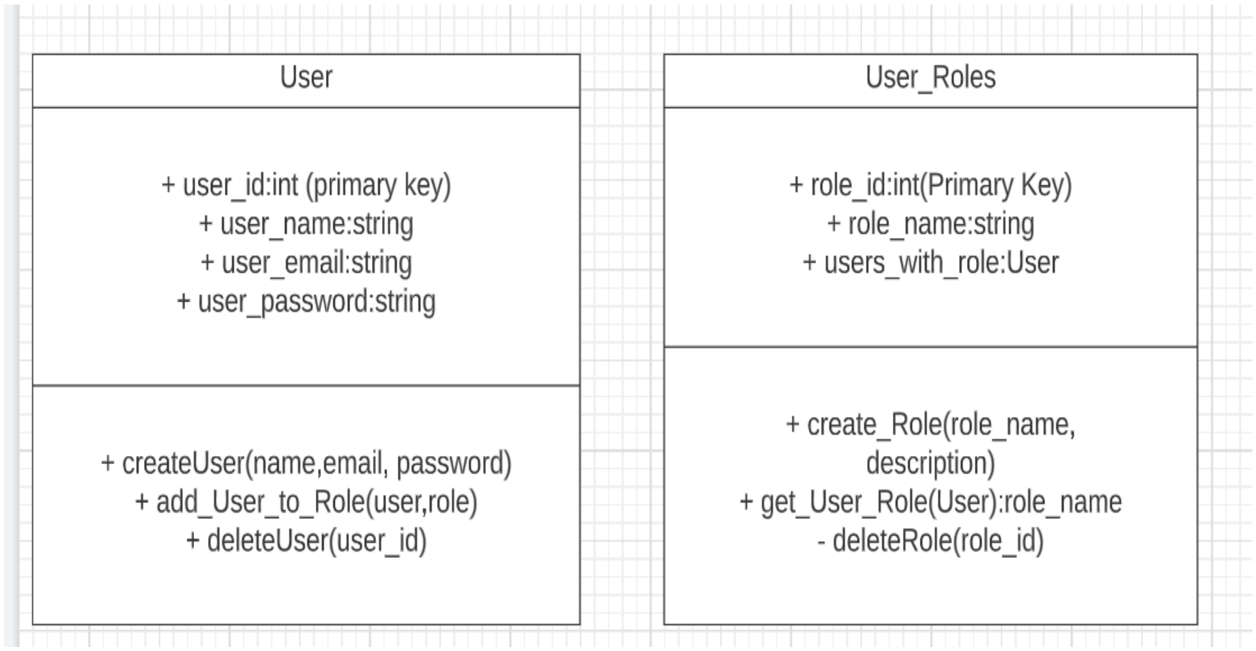

Start
Receive a Ticket
Resolve the Ticket
Check if many people are asking it
YES
Add the ticket's Query to FAQ's
NO
End

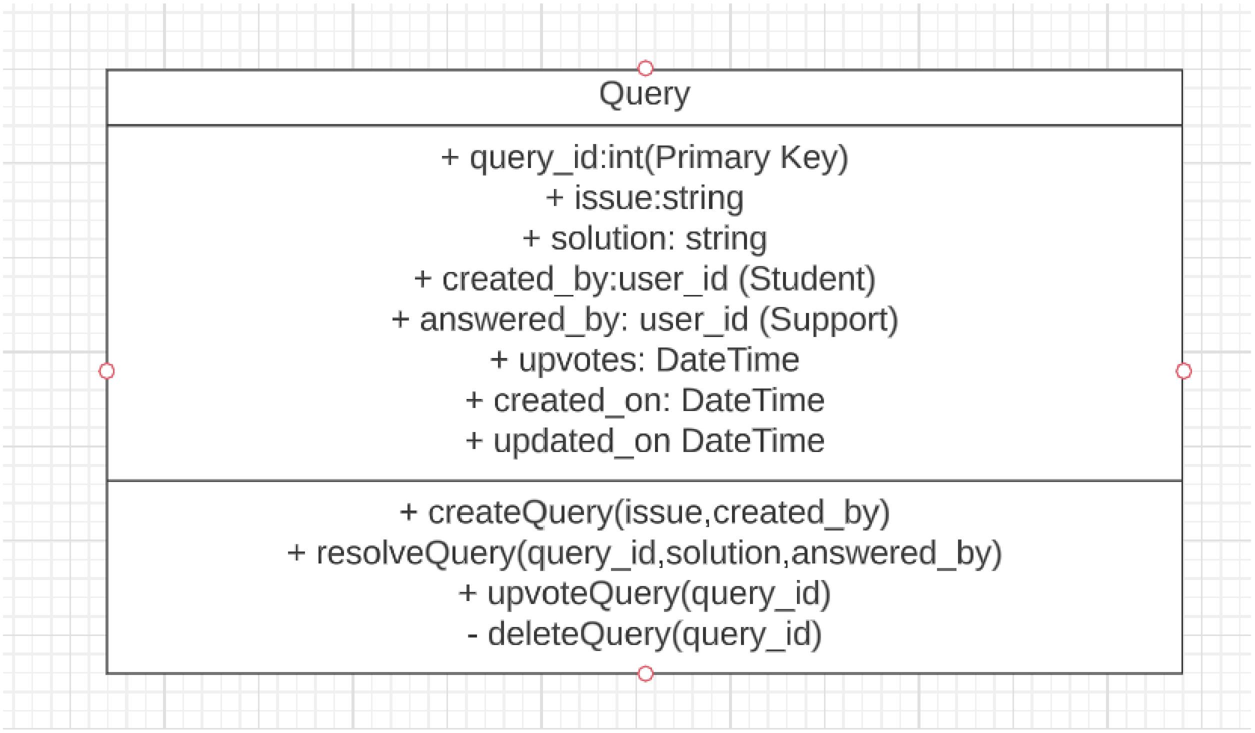
Support Team User Flow



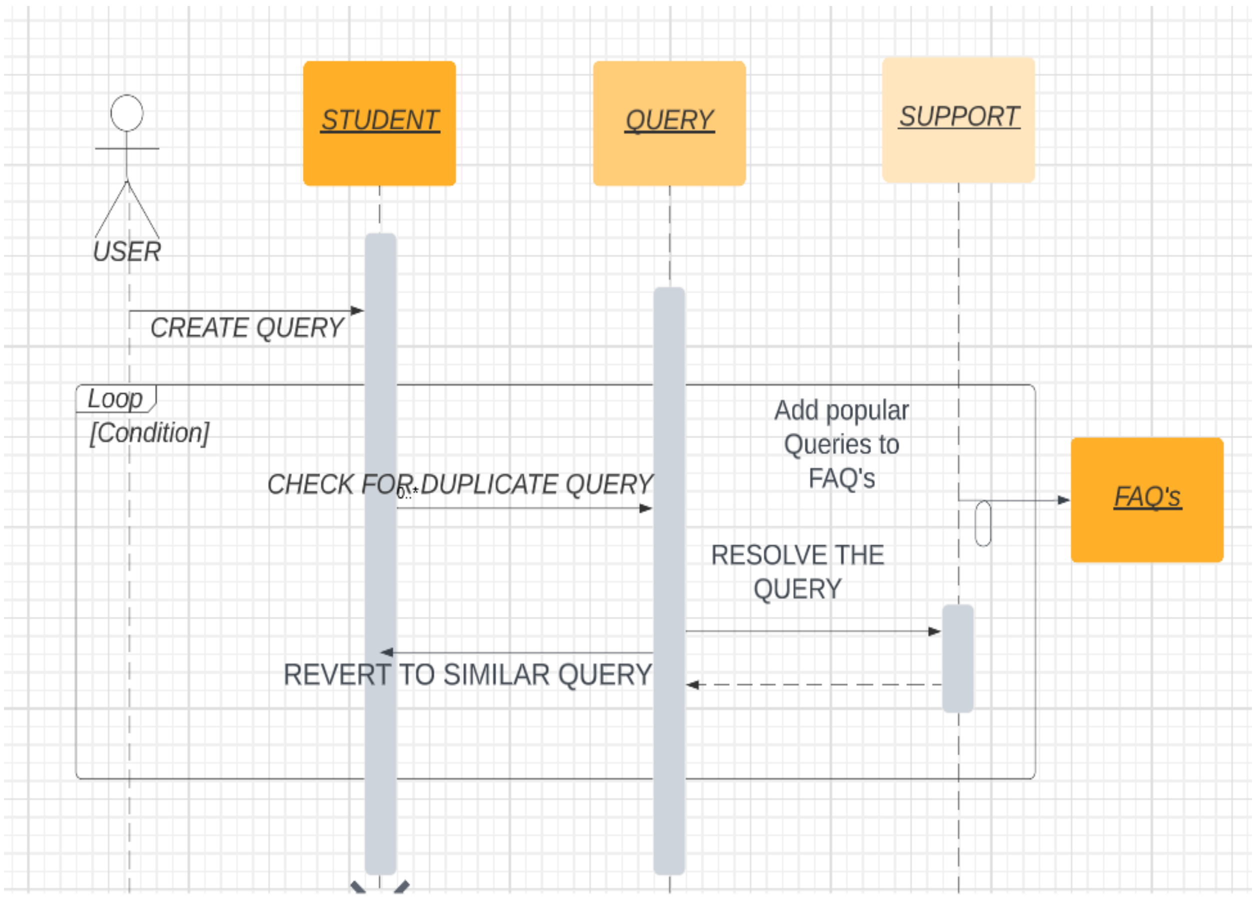
Admin User Flow

Class Diagrams (UML)


User
+ user_id:int (primary key)
+ user_name:string
+ user_email:string
+ user_password:string
+ createUser(name,email, password)
+ add_User_to_Role(user,role)
+ deleteUser(user_id)
User_Roles
+ role_id:int(Primary Key)
+ role_name:string
+ users_with_role:User
+ create_Role(role_name, description)
+ get_User_Role(User):role_name
- deleteRole(role_id)

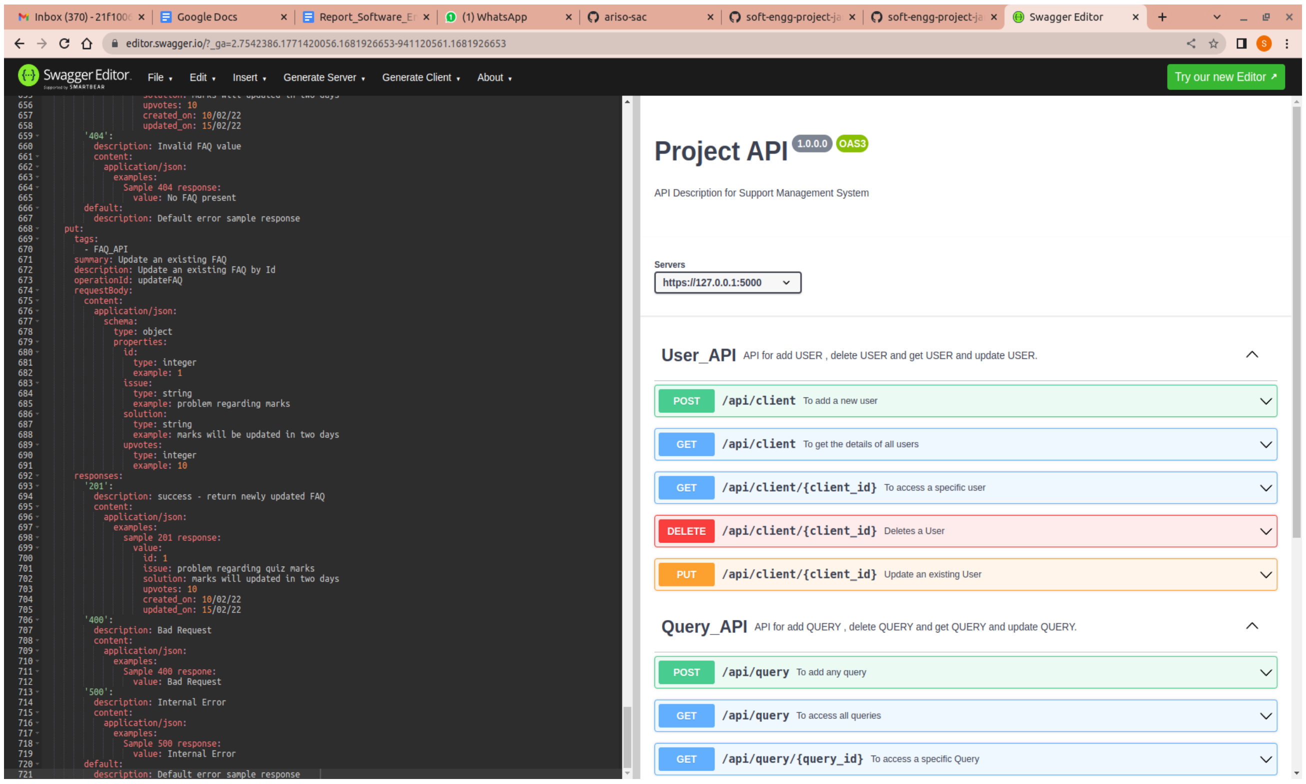

Query
+ query_id:int(Primary Key)
+ issue:string
+ solution: string
+ created_by:user_id (Student)
+ answered_by: user_id (Support)
+ upvotes: DateTime
+ created_on: DateTime
+ updated_on DateTime
+ createQuery(issue,created_by)
+ resolveQuery(query_id,solution,answered_by)
+ upvoteQuery(query_id)
- deleteQuery(query_id)

SEQUENCE DIAGRAM


STUDENT
QUERY
SUPPORT
USER
CREATE QUERY
Loop
[Condition]
Add popular Queries to FAQ's
FAQ's
CHECK FOR DUPLICATE QUERY
0..*
RESOLVE THE QUERY
REVERT TO SIMILAR QUERY

MILESTONE 4

SWAGGER UI BASED YAML FILE .



```yaml
                    solution: marks will updated in two days
                    upvotes: 10
                    created_on: 10/02/22
                    updated_on: 15/02/22
        '404':
          description: Invalid FAQ value
          content:
            application/json:
              examples:
                Sample 404 response:
                  value: No FAQ present
        default:
          description: Default error sample response
    put:
      tags:
        - FAQ_API
      summary: Update an existing FAQ
      description: Update an existing FAQ by Id
      operationId: updateFAQ
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                id:
                  type: integer
                  example: 1
                issue:
                  type: string
                  example: problem regarding marks
                solution:
                  type: string
                  example: marks will be updated in two days
                upvotes:
                  type: integer
                  example: 10
      responses:
        '201':
          description: success - return newly updated FAQ
          content:
            application/json:
              examples:
                sample 201 response:
                  value:
                    id: 1
                    issue: problem regarding quiz marks
                    solution: marks will updated in two days
                    upvotes: 10
                    created_on: 10/02/22
                    updated_on: 15/02/22
        '400':
          description: Bad Request
          content:
            application/json:
              examples:
                Sample 400 respone:
                  value: Bad Request
        '500':
          description: Internal Error
          content:
            application/json:
              examples:
                Sample 500 response:
                  value: Internal Error
        default:
          description: Default error sample response
```

Project API 1.0.0 OAS3

API Description for Support Management System

Servers
https://127.0.0.1:5000

User_API API for add USER , delete USER and get USER and update USER.

POST /api/client To add a new user

GET /api/client To get the details of all users

GET /api/client/{client_id} To access a specific user

DELETE /api/client/{client_id} Deletes a User

PUT /api/client/{client_id} Update an existing User

Query_API API for add QUERY , delete QUERY and get QUERY and update QUERY.

POST /api/query To add any query

GET /api/query To access all queries

GET /api/query/{query_id} To access a specific Query