

Master Team Project WiSe 2021

Fuldemy – Your learning help

Team 02 - Milestone 2

Dated: 14/12/2021

Team - Local

1. Kritika Murugan – Team Lead, GitHub Master
2. Debarati Ray – Backend Lead
3. Dipesh Chaudhary – Frontend Lead, M2 Editor
4. Abu Sadat Md. Soyam – QA Lead, M2 Editor
5. Syed Naqi Raza Bukhari – Backend Dev
6. Mujeeb Rahman PM - Frontend Dev

Contact:

Kritika Murugan – kritika.murugan@informatik.hs-fulda.de

Task	Task/Feedback	Date Submitted	Date Revised

1. Functional Requirements

Personae and main Use Cases

The following summarizes the two main types of personae and describes the main use cases of Fuldemy application

- **Student (Learner)**

Header

Name: Peter John

Summary quote: “Wanted to take tuition in those subjects in which he feels that he is not able to catch up”

General Characteristics

Personal Background

- Age: 25
- Gender: Male
- Marital Status: Married
- University: Hochschule Fulda
- Course: GDSD
- Education: Master

User Environment

- **Most of the time:** Busy with job and family
- **Devices:** Laptop

Psychographics

- Peter wanted to invest less time in revising concepts related to one course
- He wanted to get an expert opinion on perplexing topics in a course
- He wanted to get hands-on experience along with learning the concepts

Pain Points

- It is difficult to grasp the concepts taught in class and cover it all by himself after school hours
- It is cumbersome to find the right resources for a stubborn topic on the internet when preparing for exams
- A lack of guidance throughout the course is like finding his path through darkness

End Goal(s)

Goals	Motivators
To clear the concepts	To learn from an expert which solves queries regarding a particular topic in a course
To invest the least time on learning a new concept	To spend the least time possible in learning a new concept
To solve practice questions	To solve exercises in preparation for exams and discuss the solutions with an expert

Scenario

I am married and am parallelly working and studying. At times, it is difficult to make time for studies let alone search for the appropriate resources over the internet for perplexing topics. Even if I do find them, I still have never-ending queries in my mind. Hence, an affordably one-to-one tutoring service that would save me time would be indispensable. Having an expert opinion on certain ambiguous topics would be paramount in my learning.

- **Tutor (Expert in a specific field)**

Header

Name: Alexander Mark

Summary quote: "I would love to offer private tutoring sessions to struggling students while earning some bucks"

Demographic Information

Personal Background

- Age: 27
- Gender: Male
- Marital Status: Single
- University: Hochschule Fulda
- Education: Master's in computer science

Professional Background

Job: Full-time job as a Machine Learning researcher in Hochschule Fulda

User Environment

- **Most of the time:** At work
- **Devices:** Laptops

Psychographics

- Alexander wanted to earn some extra money while providing online tutoring

- He has extra time in the office in which he wanted to do something productive

Pain Points

- Alexander is unable to find a suitable website which provides tutoring services on school or regional level
- In available tutoring services, the competition is very high, and it is fruitless to attract students or customers
- Alexander needs a platform at the university level

End Goal(s)

Goals	Motivators
To earn extra money	To give lectures to students in his free time during the office, and to earn extra money

Scenario

Alexander is an expert in Machine Learning, and he knows exactly how a beginner struggles in this subject. Since he knows the basics very well and he has enough time in his office hours, he wants to extend his love of teaching and earn some extra money as a bonus. He is interested in giving personal sessions to students of Hochschule Fulda who are struggling in this course.

- **Admin**

Header

Name: Junaid

Summary Quote: "I want to manage the content displayed on the website in a very simple manner"

Demographic Information

Personal Background

- Age: 27
- Gender: Male
- Marital Status: Single
- University: Hochschule Fulda
- Education: Master's in computer science

Professional Background

Job: Full-time job in the admin office at Fulda University of Applied Science

User Environment

- **Most of the time:** in the office
- **Devices:** Laptops

Psychographics

- Junaid is very good at filtering content on websites, as he has been in this job for 5 years
- He is currently working in the admin office and currently also responsible for the content approval of the Hochschule Website

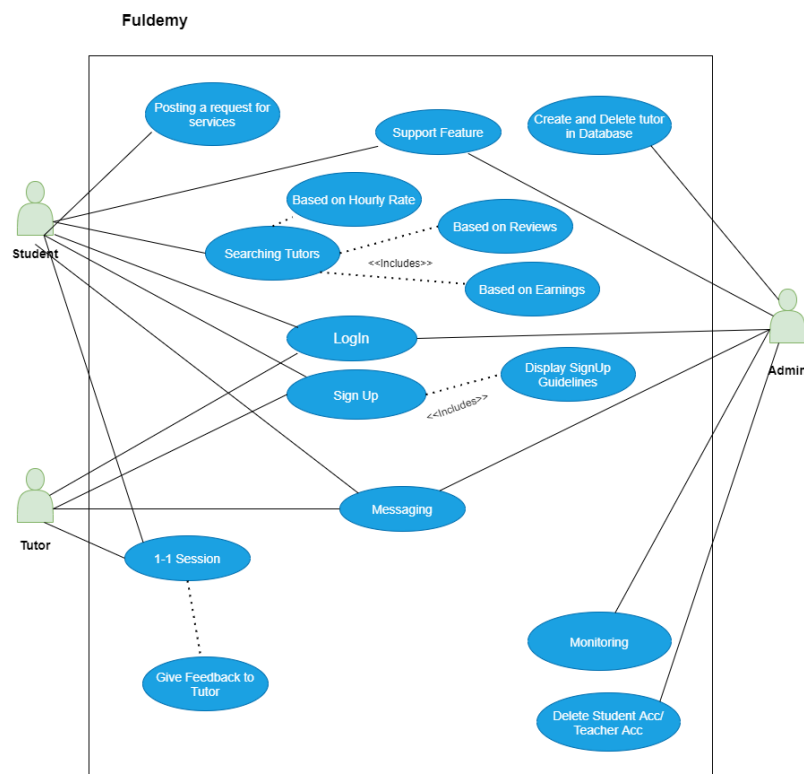
End Goal(s)

Goals	Motivators
To earn extra money	He wants to earn some extra money by providing his duties of administrator of the website

Scenario

Junaid is already working in the admin department of the university, and he is an expert in checking and approving the content which has been displayed on the Hochschule website as well as notice boards, he makes sure that something which is not acceptable for Hochschule is not displayed on notice boards or the main website. He has some free time during his job and wants to earn some money while providing administration to such a website which also needs to double-check the content that is going to upload on the website.

Use Cases



Use Case Diagram

List of use cases

- Login
- Sign Up
- Searching Tutors
- Search based on Views
- Search based on Earnings
- Search based on the hourly rate
- Posting a request by a student
- Display guidelines for Sign Up Process
- Request tutoring Services
- One-to-one Session between Student and Tutor
- Give Feedback
- Support feature for student
- Create profile
- Delete profile
- Monitoring
- Messaging between Student and Tutor

Main Use Cases

- **Support Feature**

Students can benefit from the Support Feature which will be available from 8 am to 8 pm. If the students' desired teacher is not available at any particular moment, they can send the request to support staff which will answer the query at the earliest. In this way, the students need not wait for their tutor to log in and answer the query, especially when time is a concern

- **Give Feedback**

Students can give feedback to their respective teacher/tutor from whom they avail assistance. This feedback helps other students to pick tutors. This is the main tool for maintaining the quality of tutoring services offered on the website

- **Post a request for tutoring**

They can choose a specific subject in which they need help and post a request on the platform. This request will be live once the admin approves it. This is necessary to ensure that students do not post inappropriate or duplicate content.

- **Messaging**

The in-site messaging tool facilitates the interaction between students and tutors without the need for third party tools. The conversation will be saved and visible at any time.

- **Monitoring & Approval**

The admin would monitor the changes in the profile of the tutor if their profile is approved. For example, if the tutor updates their CV, the admin must first approve the changes before they are made public. This use case is very important to maintain and allow only appropriate content on the website.

Functional Requirements

Browse Tutors (Use Case 1)

1. Students should be able to click on the displayed categories from the navigation bar
2. They can click on any specific category (subjects)
3. Students should be redirected to a selected category of the tutors page
4. Students can check tutors on that page and can contact any tutor that is online

Tutor Search

1. Students can input the search keywords in the search box
2. The search button is clickable
3. Students should be redirected to the search result page with correct search results
4. Students can change the search criteria from navigation bar and search again

Search Tutors based on Earnings and Hourly Rate (Use Case 2 & 3)

1. Users should be able to select the search category (Earnings & Hourly Rate)
2. Users can select one option from the search options
3. Users will be redirected to the filtered results
4. Users can check tutors on that page and can contact the tutor that is already logged in

Sign up/ Registration (Use Case 7 & 8)

1. Users can click on the Sign up button, and they will be redirected to a page containing guidelines of the Sign In process
2. A user can register in the system using "Fulda" or "SFSU" email address and a password with instructed restrictions
3. A new user will be asked if they want to register as a student or tutor
4. If an existing user tries to sign up again, an error message "User already exists" should be clearly displayed
5. If the user's password strength does not match the system. A detailed password guide should be displayed
6. Users must agree with data privacy while signing up

Sign In (Use Case 6)

1. A user should be able to sign in successfully in the system if the provided user email & password is correct
2. If the user's details are wrong, the system should render "Email or Password is not valid"

Posting Request (Use Case 5)

1. Students can post a request if they are logged in to the account
2. Students can see the previously submitted requests from their account

Support Functionality (Use Case 12)

1. Students can click on the support button which is appearing on their dashboard if they are subscribed to some tutor and are taking classes from them. The students can ask any queries from the support team member regarding subscribed course
2. A support team member will help the student when the tutor is offline

Feedback for Tutors (Use Case 11)

1. Students can give feedback ranging from 1 to 5 stars to tutors
2. Feedback option will be available when the course is completed with a tutor
3. Tutors will be ranked according to ranking when students search for tutors

Dispute with Tutor (Use Case 19)

1. Students can file a grievance if they are not satisfied with the services offered by the tutor
2. A button will appear on the student dashboard on which they can click and file their grievances
3. Students will explain the matter in the form and then click the submit button to
4. The administrator will review the submitted form and take appropriate action

A timetable of Tutor (Use Case 18)

1. Students can click on the button named "**Timetable**" which will be available on the tutor's profile.
2. Students then see the slots of when the tutor would be available for live chat

Create & Delete Profile (Use Case 13 & 14)

1. An administrator can create or delete a profile of a student or tutor

Monitoring & Approval (Use Case 15)

1. Whenever tutor registers on the website, the profile will be checked by the administrator, and then they will decide whether to approve or reject the profile

2. If the tutor edits their profile after approval, then it will again be submitted for a review, and the admin will manually approve or disapprove new changes

Chat Feature (Use Case 16)

1. After the student has chosen a tutor, the chat functionality will be available for them

#	User Type	Feature	Priority
1	Student	Browse Tutors	1
2		Search for Tutor based on Hourly Rate	1
3		Search from Key Word	1
4		Search for Tutor based on Earnings	1
5	Student + Teacher	Sign Up /Registration	1
6		Login	1
7	Student	Posting Request	1
8		Feedback of Tutor	1
9	Admin	Student profile creation	1
10		Tutor profile creation	1
11		Tutor profile approval	1
12	Student	Support Functionality	1
13	Student + Tutor	Display Guidelines for Sign Up Process	1
14		Promotional Emails and Newsletter	1
15	Admin	Can delete student/teacher account	1
16	Student + Admin	Dispute with Tutor	2
17	Student + Tutor	In Messaging Chat Feature	1
18	Student	Timetable of tutor	2
19	Student + Tutor	1-1 Session with Student and Tutor	2
20	Teacher	Teacher can update their timetable	2
21	Student + Tutor	Video Call Feature	3
22		Bug or Error reporting	3

2. List of main data items and entities

Entity List	Purpose of the Entity	Data Structure
Students	This entity will be used for keeping all attributes of the students who will register to the website for learning and growth.	Table
Tutors	This entity will be used for keeping all attributes of the tutors who will register to the website for giving guidance and tutorials.	Table
Admins	This entity will be used to keep track of the admin related attributes and other details.	Table
Skills	This entity will be used to keep information about all the subjects that will be taught by tutors and related attributes to it	Table
Active_Classes	This entity will be used to keep track of ongoing classes linking students, tutors, skills and the admin monitoring it.	Table
Inactive_Classes	This entity will be used to keep track of historical classes linking students, tutors, skills and the admin monitoring it.	Table
Support	This entity will be used to keep track of all the support related queries asked by student	Table

ENTITIES

Entity Name: Admins

Name of the Column	Constraints if any	Description	Data Type
Admin_id	PK	Unique Surrogate key to be generated by the system for the admin.	int
Admin_first_name	NOT NULL	Name of the admin.	varchar(30)
Admin_last_name		Last Name of the admin	varchar(30)
Admin_birth_date		Birthdate of admin	date
Admin_password		Password to login.	varchar(30)
Admin_email	Unique	Email of the admin	varchar(255)

Entity Name: Skills

Name of the Column	Constraints if any	Description	Data Type
Skill_id	PK	Unique Surrogate key to be generated by the system for all the skills that are decided by the admin to be taught in Fuldemy	int
Skill_name	NOT NULL	Name of the skill. E.g: "Machine learning for Beginners", "Maths 101"	varchar(30)
Skill_level		This column would determine the level of the skill to be taught.	varchar(30)
Skill_admin	NOT NULL	Admin id who has entered the skill details in the table.	int
Skill_description		Complete Details of the skills topics to be covered	varchar(255)
Skill_department		Department Name of the Skills. E.g: Applied Computer Science	varchar(30)

Skill_duration_days		Skill duration to complete the course	int
---------------------	--	---------------------------------------	-----

Entity Name: Students

Name of the Column	Constraints if any	Description	Data Type
Student_id	PK	Unique Surrogate key to be generated by the system for all the student registering in the system	int
Student_first_name	NOT NULL	First name of the student	varchar(30)
Student_last_name		Last Name of the students	varchar(30)
Student_address		Address of Student	varchar(100)
Student_email		Fulda Hochschule id of the student Unique constraint should be present in this column	varchar(255)
Student_registration_date		Date to register in the system	date
Student_profile_picture		Picture of the student	blob
Student_phone_number		Contact Details	bigint
Student_password	NOT NULL	Encrypted password of the student	varchar(30)
Student_interests		Skills Id in comma separated order	varchar(100)

Entity Name: Tutors

Name of the Column	Constraints if any	Description	Data Type
Tutor_id	PK	Unique Surrogate key to be generated by the system for all the Tutor registering in the system	int
Tutor_first_name	NOT NULL	First name of the Tutor	varchar (30)
Tutor_last_name		Last Name of the Tutor	varchar(30)
Tutor_address		Full Address of Tutor	varchar(100)
Tutor_email		Fulda Hochschule id of the student Unique constraint should be present in this column	varchar(255)
Tutor_registration_date		Date to register in the system	date
Tutor_profile_picture		Picture of the student	blob
Tutor_phone_number		Contact Details	bigint
Tutor_password	NOT NULL	Encrypted password of the Tutor	varchar (30)
Skills		Skills Id in comma separated order	varchar (30)
Hours_weekly		Weekly bracket one can teach	int
Tutor_is_active		Admin will change to true if the tutor is approved.For any change in CV, it will automatically change it to false	boolean

Entity Name: Active_Classes

Name of the Column	Constraints if any	Description	Data Type
Class_id	PK	Unique Surrogate key to be generated by the system for all the student registering in the system	int
Student_id	FK	Student id of the tutor who would take the class	int
Tutor_id	FK	Tutor id of the tutor who would take the active class	int
Skill_id	FK	Skill id of the Skill table	int
Admin_id	FK	Admin of the class	int
Class_start_date		Class registration date	date
Skills_duration_left		Skill duration left in days	int
Class_description		Class course details	varchar (255)
Rating_by_student		Rating of the tutor given by the student based on the class taken.	int
Feedback_in_words		Feedback of the student in written words given based on her/his experience	text
Payment_type		Payment method to be done by student	Varchar(10)

Entity Name: Inactive_Classes

Name of the Column	Description	Data Type
Class_id	Unique Surrogate key to be generated by the system for all the student registering in the system	int
Student_id	Student id of the tutor who would take the class	int
Tutor_id	Tutor id of the tutor who would take the active class	int
Skill_id	Skill id of the Skill table	int
Admin_id	Admin of the class	int
Class_start_date	Class registration date	date
Class_end_date	Class registration end date	date
Skills_duration_left	Skill duration left in days Default value is 0	int
Class_description	Class course details	varchar(255)
Rating_by_student	Rating of the tutor given by the student based on the class taken.	int
Feedback_in_words	Feedback of the student in written words given based on her/his experience	text
Payment_type	Payment method to be done by student	Varchar(10)

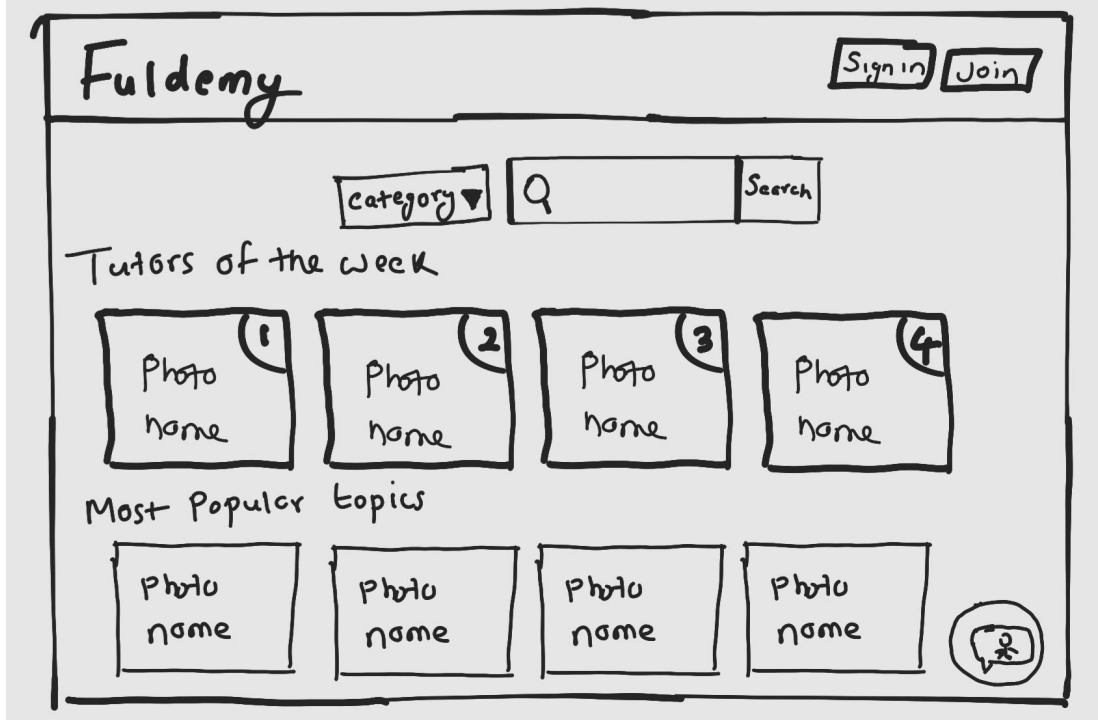
Entity Name: Support

Name of the Column	Constraints if any	Description	Data Type
Support_id	PK	Unique Surrogate key to be generated by the system for all the student asking for support in the system	int
Support_status		Current Status of the query asked. Resolved Waiting for a Reply	Varchar(10)
Student_id		Student id of the student who asked the query.	int
Admin_id		Admin of the class	int
Support_tutor_id		Tutor id of the student who will be assigned by the system to answer the question.	int
Active_class_id	FK	Class id of the active class of the skill taken by the student	int
Support_query		Skill duration left in days Default value is 0	text
Support_attached		Attached document by the student related to the query	blob
Support_query_date		Date of Support query posted in the system	date
Tutor_comments		Reply added by the tutor.	text
Admin_comments		Admin comments if required	text

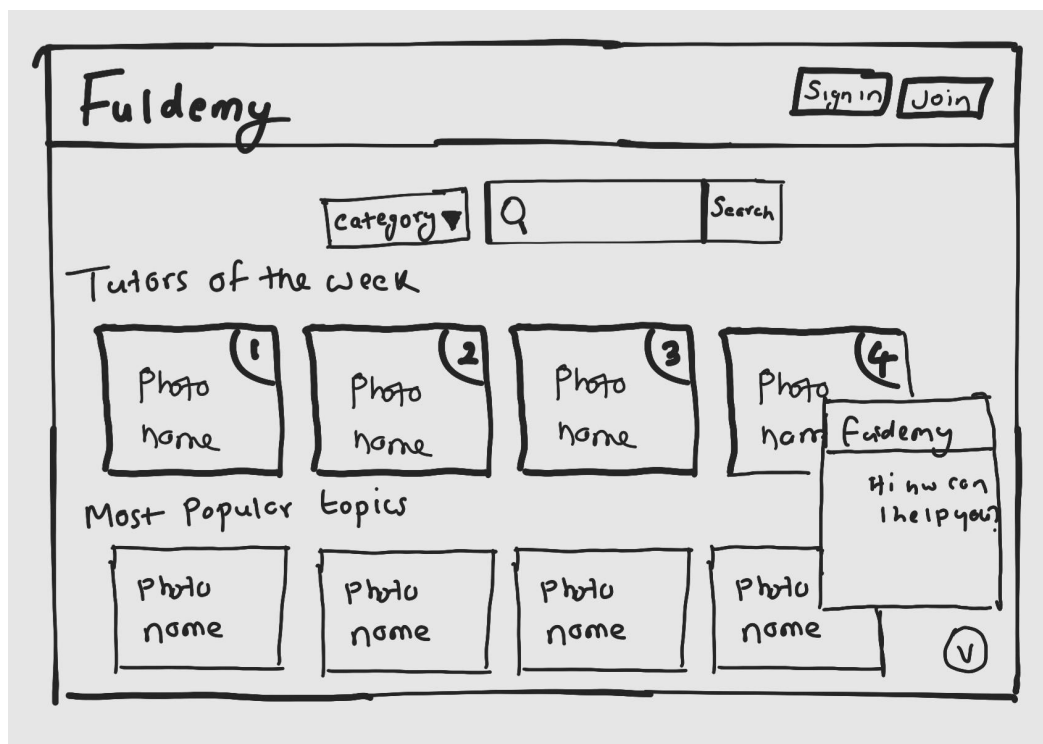
3. UI Mockups and Storyboards:

Browsing option for anyone without logging in

① Browse Tutor



when the chatbot icon is clicked



Searching by keyword Math101

The mockup shows a web interface for 'Fuldemy'. At the top, there's a header with the site name and 'Signin' and 'Join' buttons. Below the header is a search bar with a 'category' dropdown, a text input containing 'Math 101', and a 'Search' button. Under the search bar, the text 'Math 101 :' is displayed. To the right of this text are 'Filter' and 'Sort by' dropdown menus. Below these are three product cards. Each card contains a 'Profile picture' placeholder, a 'Name' field, a rating (indicated by an asterisk and a number), and a 'Starting @' price per hour. The first card has a rating of 5 and a price of €15/hr. The second card has a rating of 4.8 and a price of €13/hr. The third card has a rating of 4.5 and a price of €10/hr. Below the product cards, there are three bullet points explaining the filters and sorting options:

- * Category \Rightarrow Science, Business, Language
- * Filter \Rightarrow Language (English, Deutsch), Rating (5*/>4*/>3*)
- * Sort by \Rightarrow Price (low to high/high to low), Highest rated

To get further Tutor info, user need to join Fuldemy

③ Registration

The mockup shows a registration page titled 'Join Fuldemy'. It features three input fields for 'Email', 'Password', and 'confirm Passwrd'. Below these fields is a 'Submit' button. At the bottom of the page, there is a link that says 'Already a member? Sign In'.

② Sign In

Fuldemy

Sign In to Fuldemy

Email

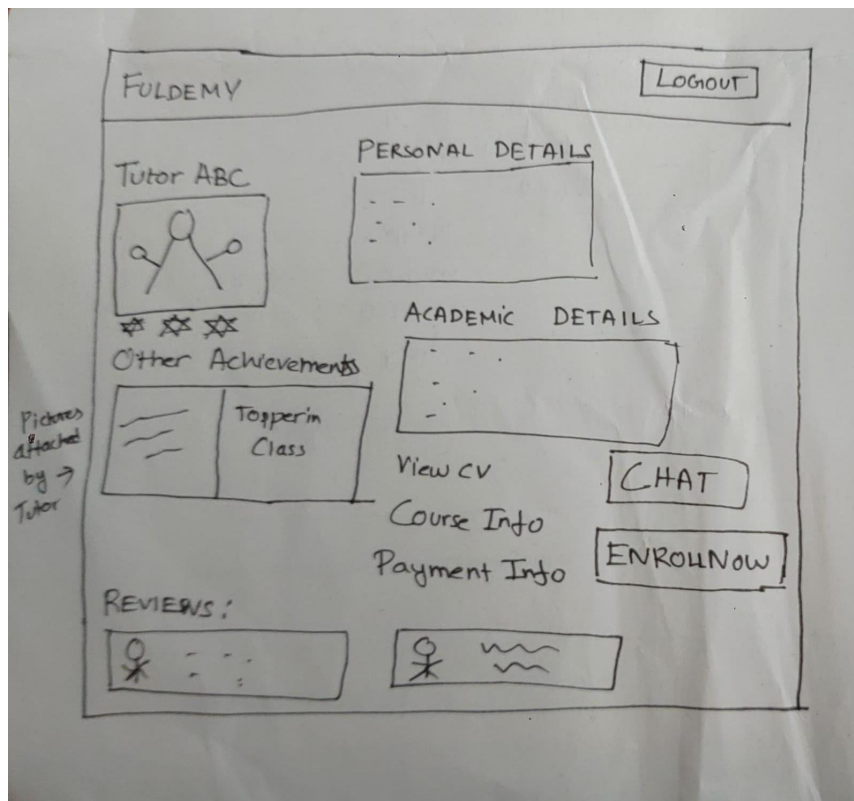
Password

Submit

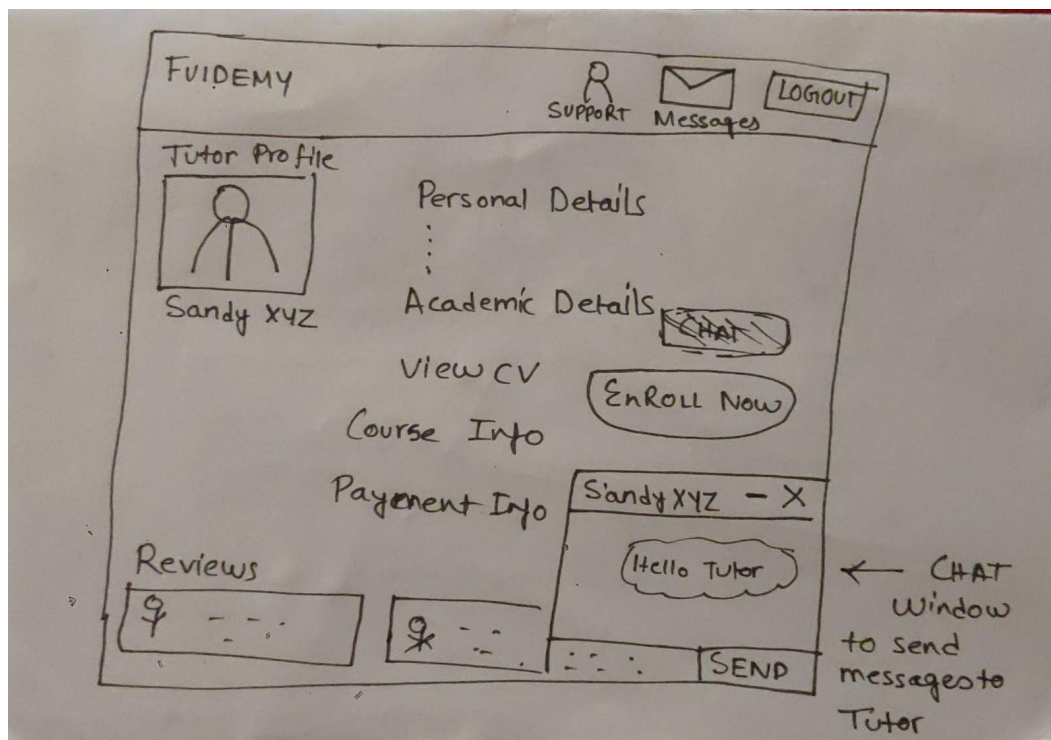
Forgot Password

Not a member yet? Join Now

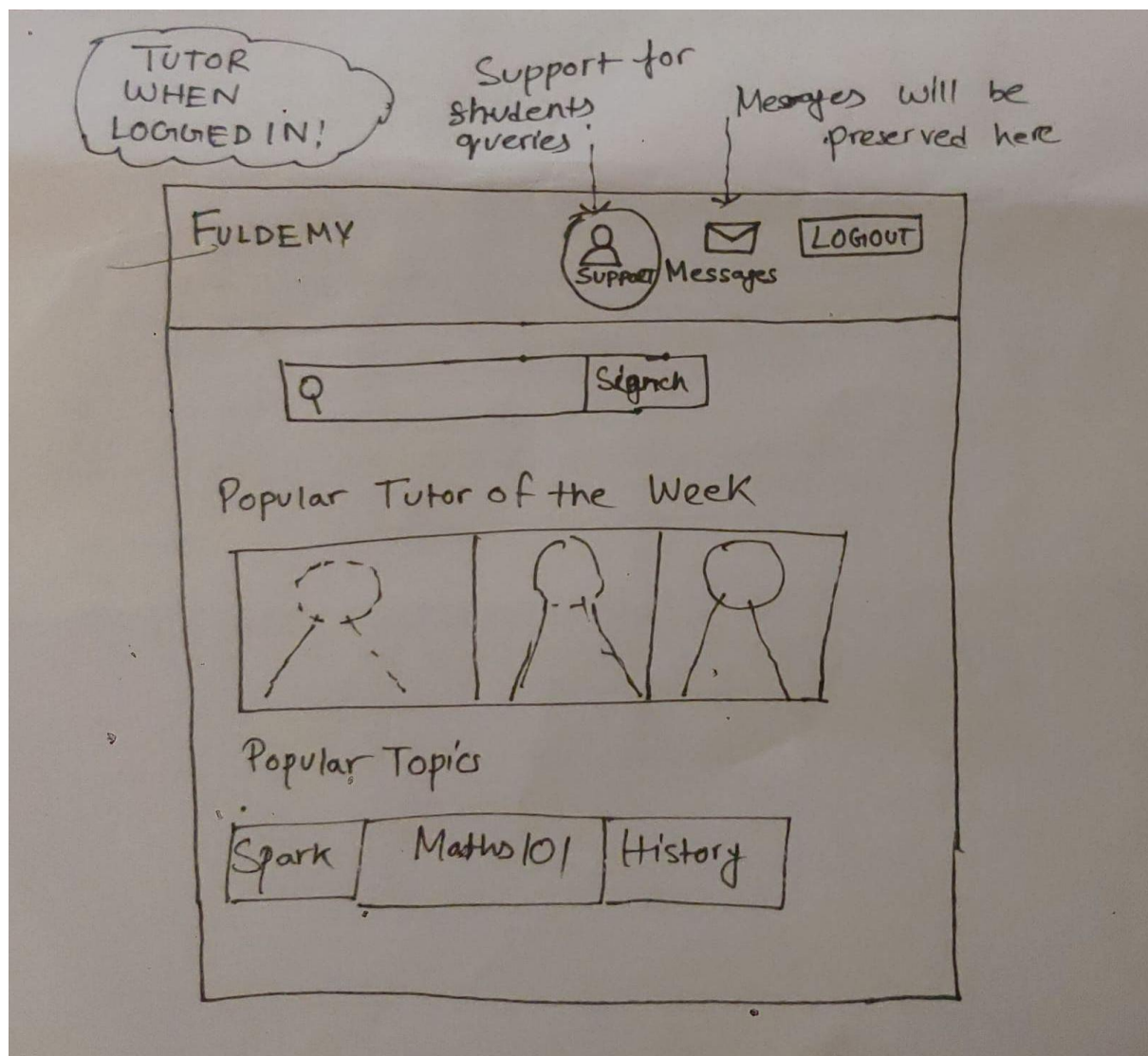
Once student registers, the tutor detail information page can be visible.



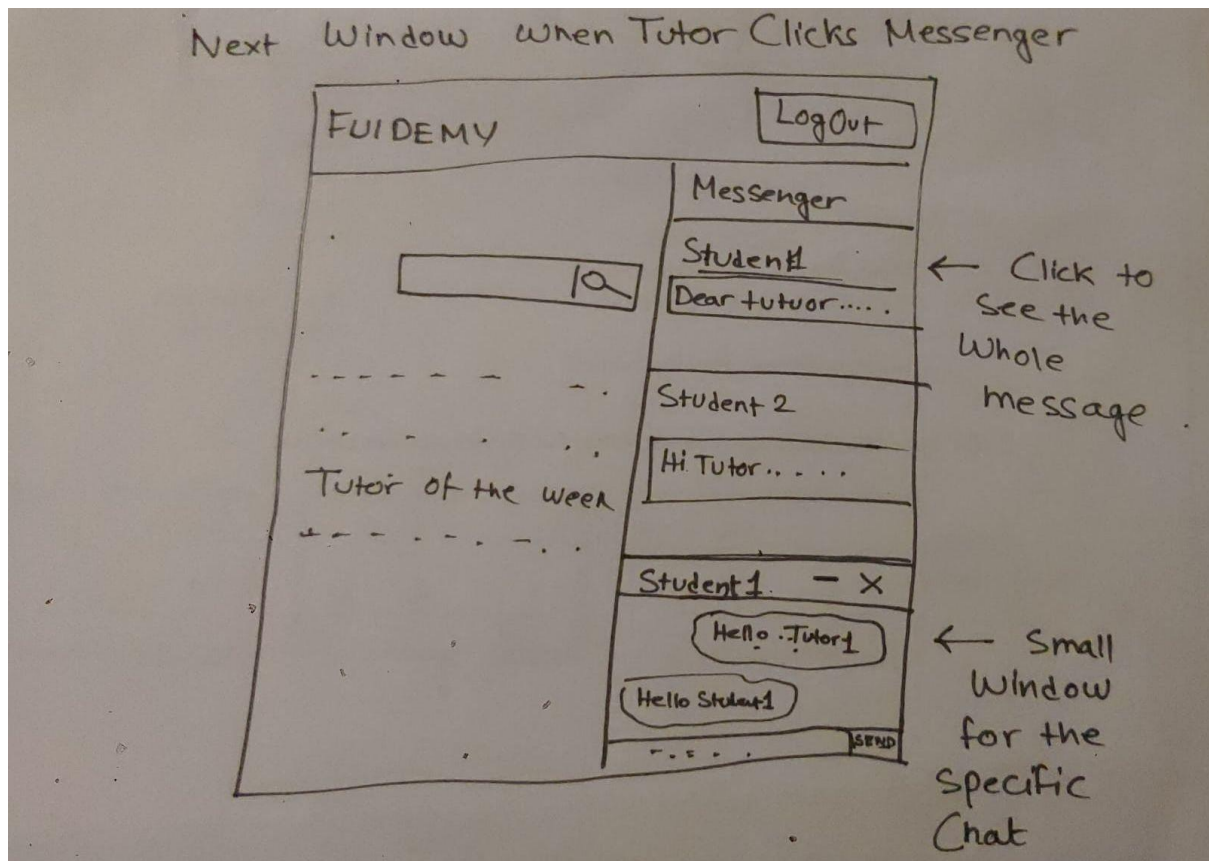
When student clicks on Chat button, small chat window opens up to send messages



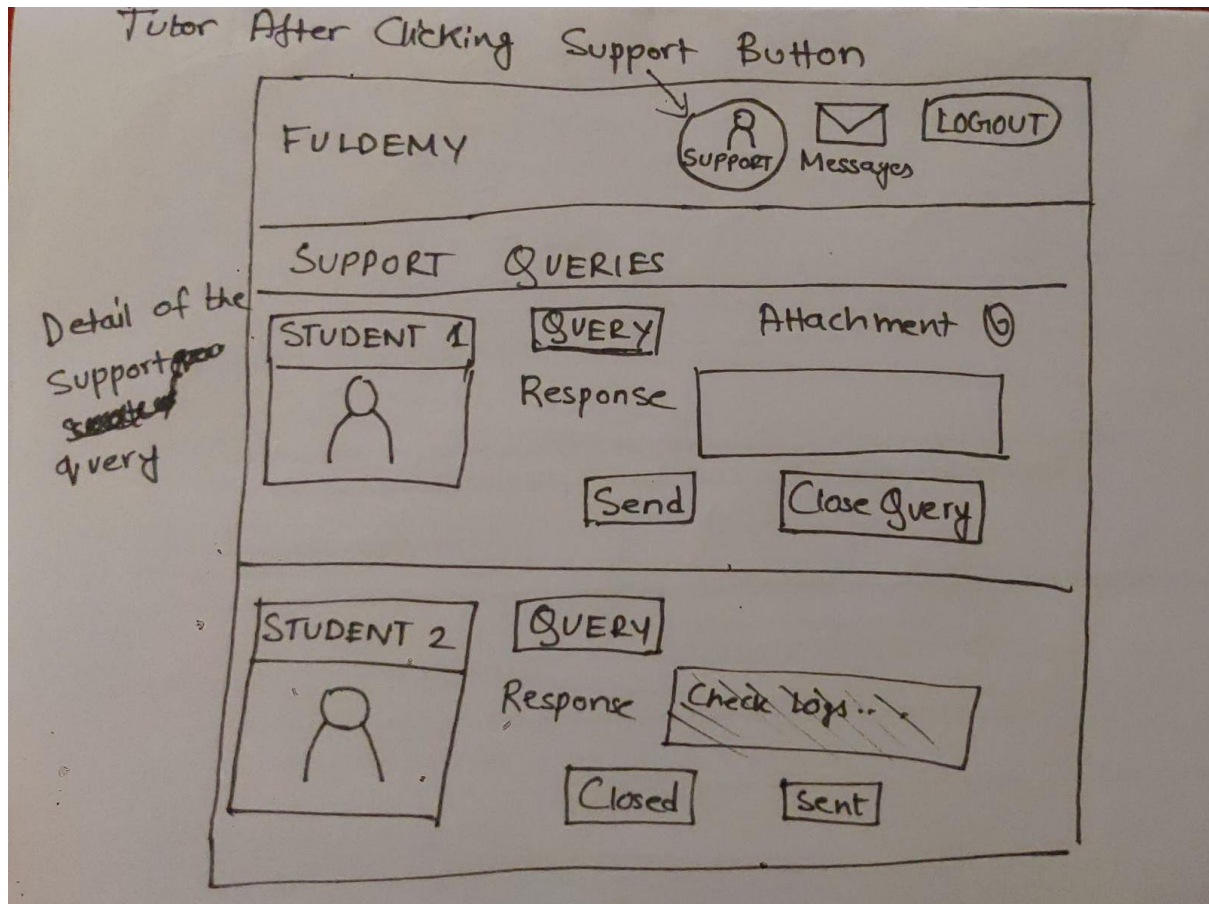
When tutor logs in the portal



Tutor can click on my Messages button to see all the messages sent by other students



Tutor can click on Support Window to see all the support Queries by students



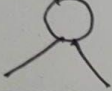
Tutor Profile Register Page

Tutor Creation Profile

Fuldemy

Logout

Tutor Profile



Change Profile

Add pictures of any achievement

+

+

First Name

Last Name

Address

Email

DOB

DD MM YYYY

Phone

Skills

Weekly Hours Available

Fees ~~Hourly~~

Upload CV

File

Save Profile

User can edit existing profile as well.

Many (2) pictures can be added for showing achievement

Fees per hour or Fees per Day per session

Student Creation Page:

Student Create Profile

Ful demy

Student Profile

First Name:

Lastname:

Email:

DOB: Date

Phone no:

Interests:

Link to change Password.

Notification

Profile.

This Page is initially not editable, view page. There will be an edit button, when user clicks on edit button, editable form (shown here) will appear.

Feedback and Rating Form for Enrolled Student:

TUTOR DETAIL PAGE

NAME:

OTHER DETAILS:

FEEDBACK:

RATING: ☐ ☐ ☐ ☐ ☐

OTHER FEEDBACKS:

List of Feed Back & Ratings

Fuldenye

Queries support [Logout](#)

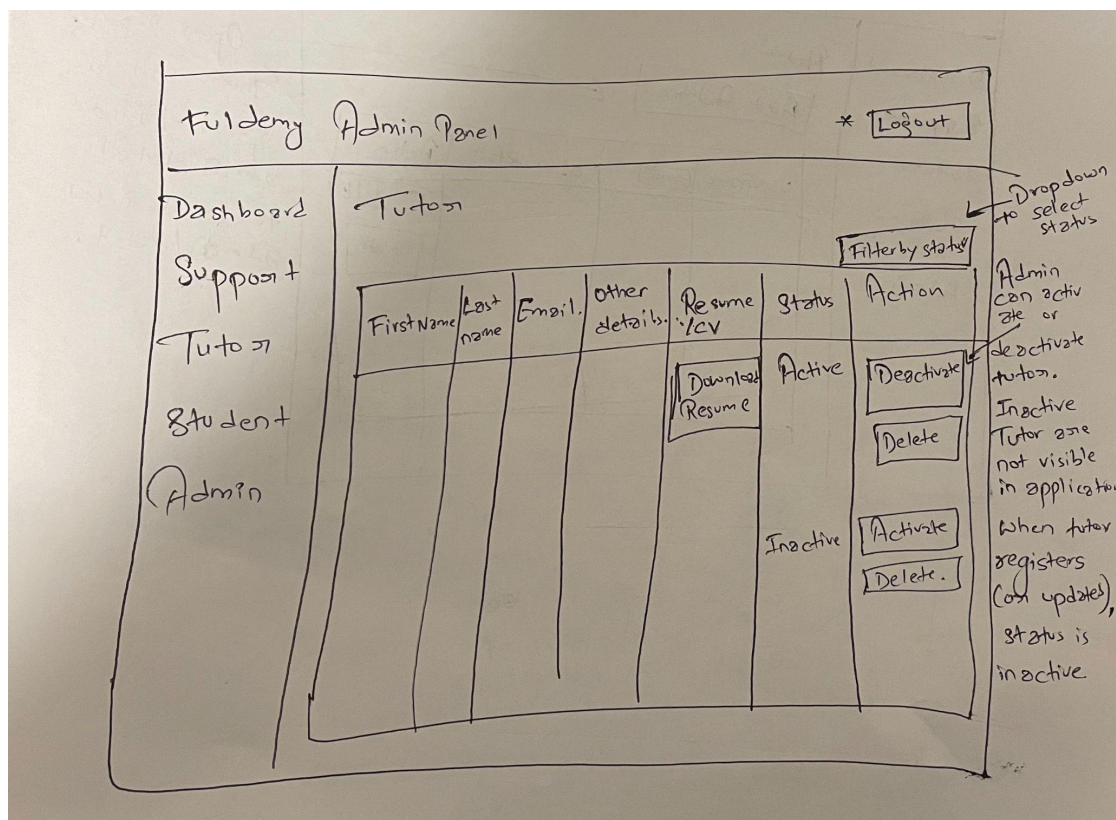
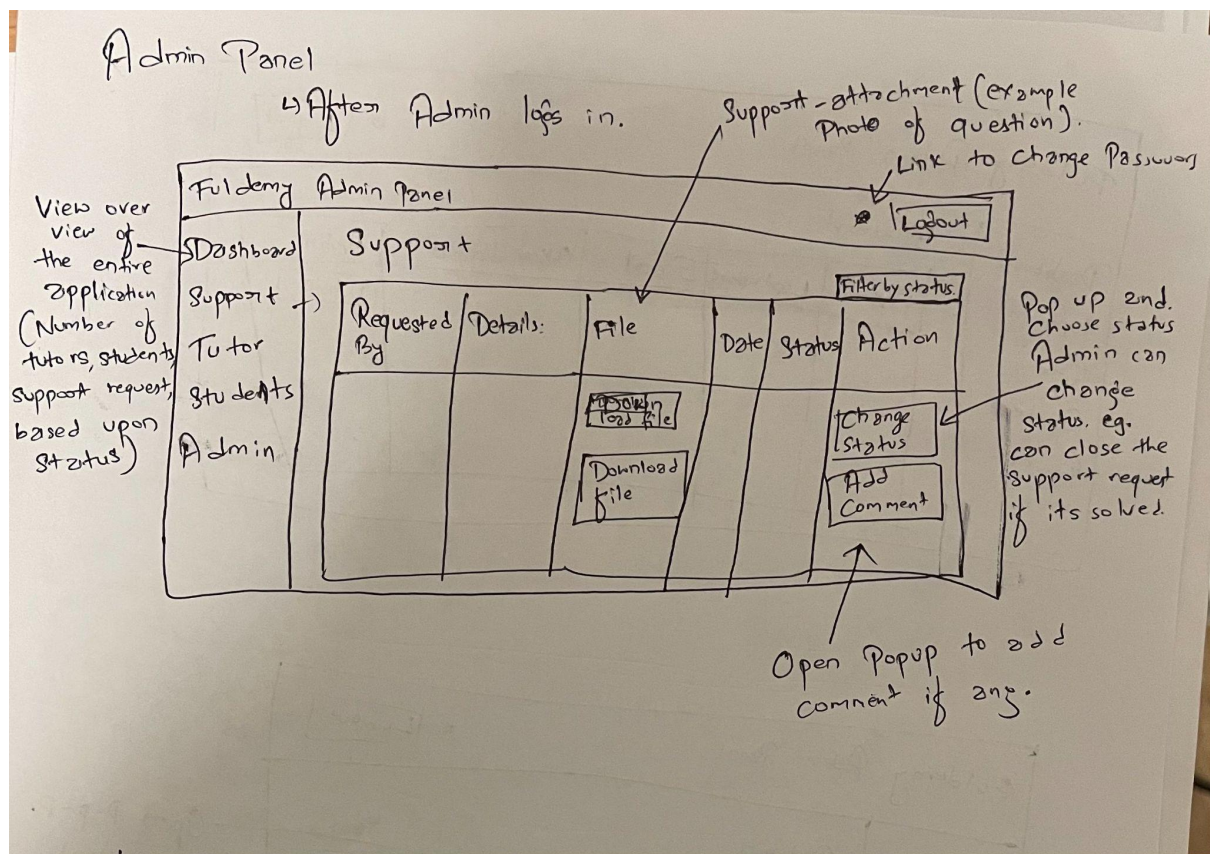
Post Query:

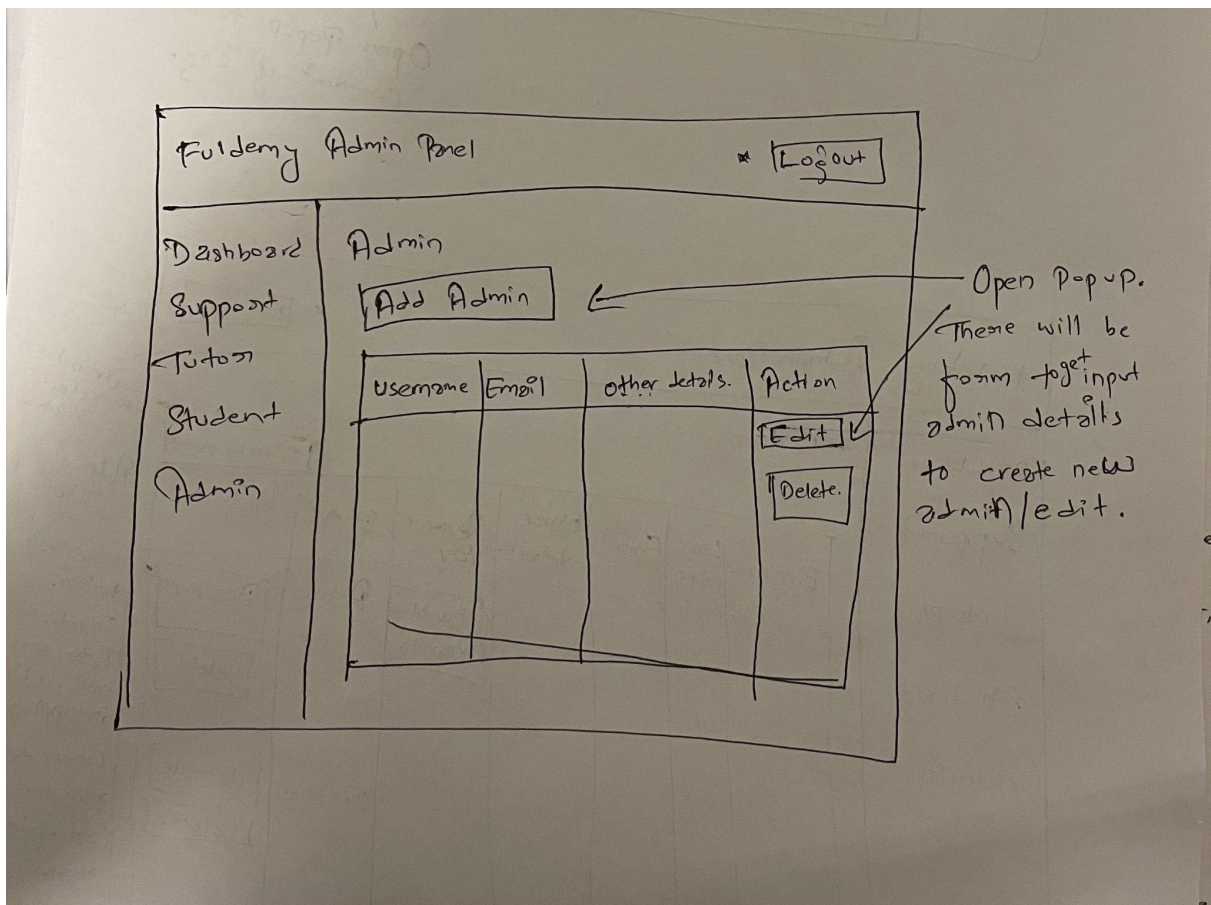
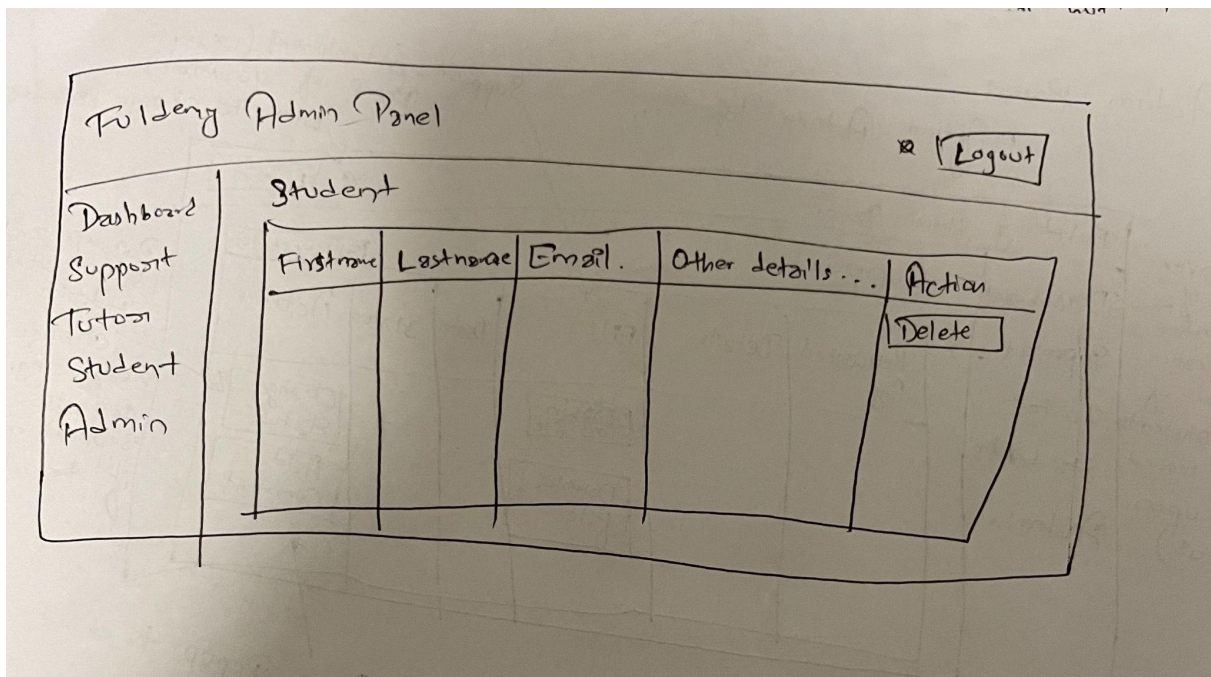
Description

Category:

Request to:

Admin Portal





4. High Level Architecture & Database Organization:

Name of the DB: Fuldemydb

Admins
admin_id int
admin_first_name varchar(30)
admin_last_name varchar (30)
admin_birth_date date
admin_password varchar (30)
admin_email varchar (255)

Tutors
tutor_id int
tutor_first_name varchar(30)
tutor_last_name varchar(30)
tutor_address varchar(100)
tutor_email varchar(255)
tutor_birth_date date
tutor_registration_date date
tutor_profile_pic blob
tutor_resume blob
tutor_phone_number bigint
tutor_password varchar(30)
skills varchar(30)
hours_weekly int
fees_daily int
tutor_is_active boolean

Students
student_id int
student_first_name varchar(30)
student_last_name varchar(30)
student_address varchar(100)
student_email varchar(255)
student_birth_date date
student_registration_date date
student_profile_pic blob
student_phone_number bigint

student_password varchar(30)
student_interests varchar(100)

Active Classes
class_id int
student_id int
tutor_id int
skill_id int
admin_id int
class_start_date
skill_duration_left int
class_description varchar(255)
rating_by_student int
feedback_in_words text
payment_type varcahar(10)

Inactive Classes
class_id int
student_id int
tutor_id int
skill_id int
admin_id int
class_start_date
class_end_date
skill_duration_left int default 0
class_description varchar(255)
rating_by_student int
feedback_in_words text
payment_type varcahar(10)

Support
support_id int
support_status varchar (10)
student_id int
admin_id int
support_tutor_id int
active_class_id int

support_query text
support_attached blob
support_query_date date
tutor_comments text
admin_comments text

Search/filter architecture and implementation:

- SQLs with %like to be used to search from the DB tables
- Filter Searches based on Topic (like Maths 101, ML Basics), highest rating of tutor, lowest price of tutor.
- Popularity Ranking will be done based on highest reviews given by students as part of feedback.

5. High Level UML Diagrams

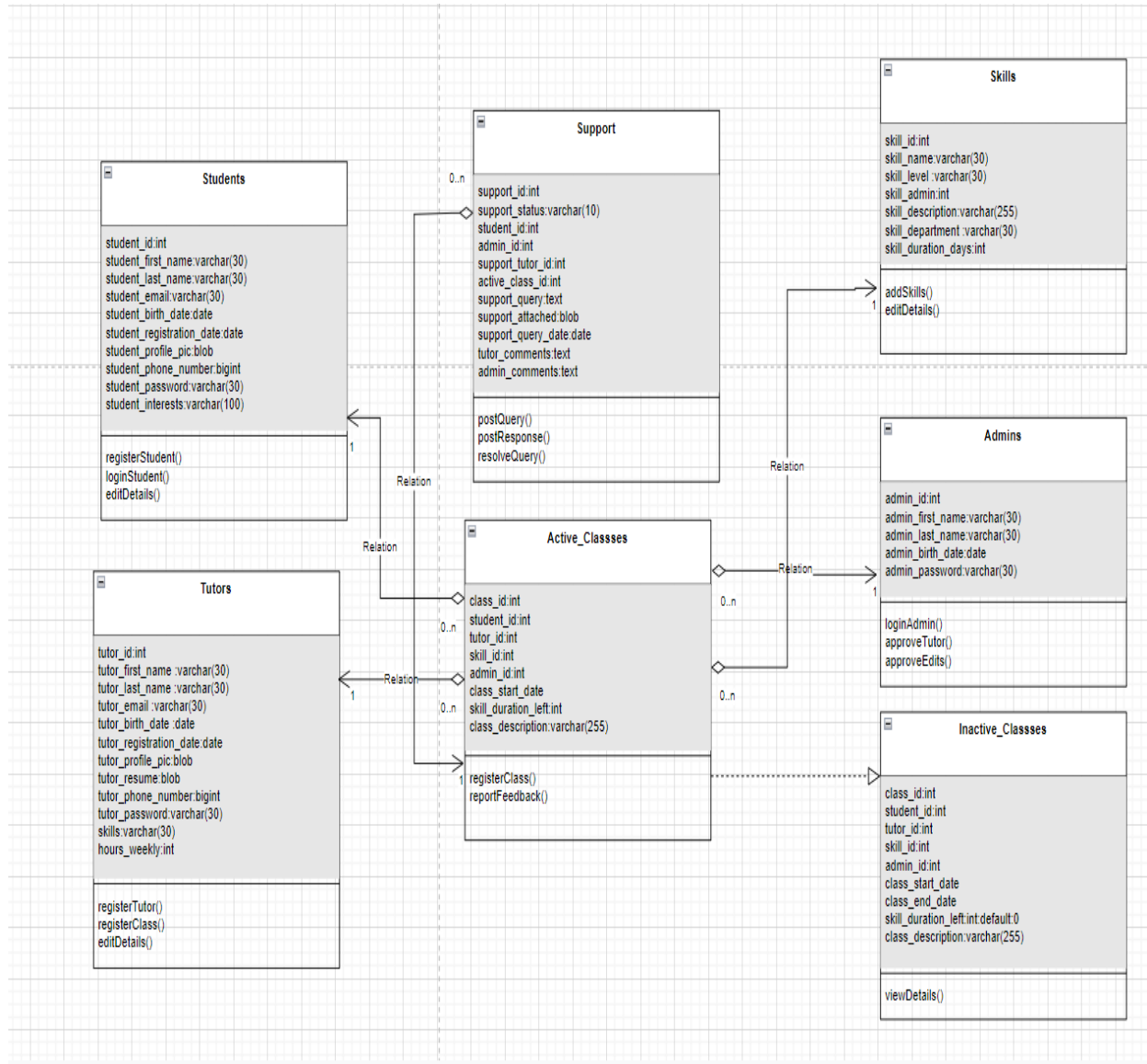


Fig. Class Diagram

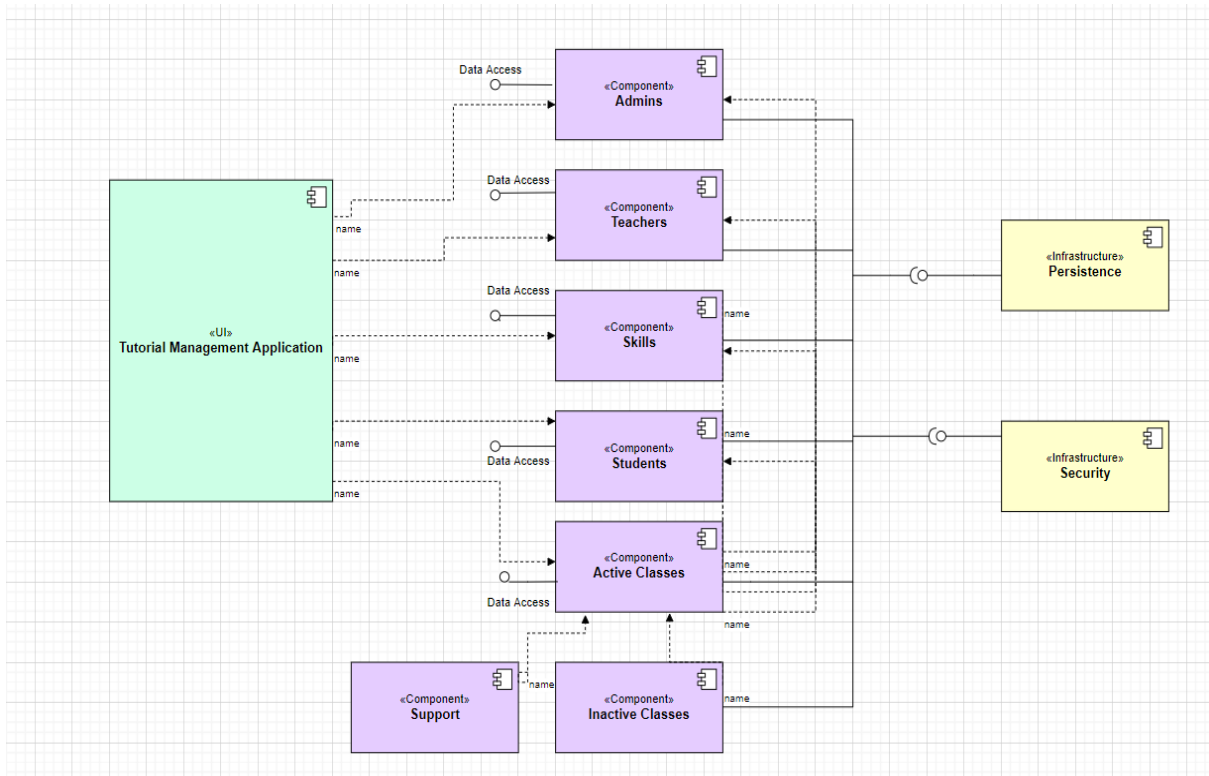


Fig. Component Diagram

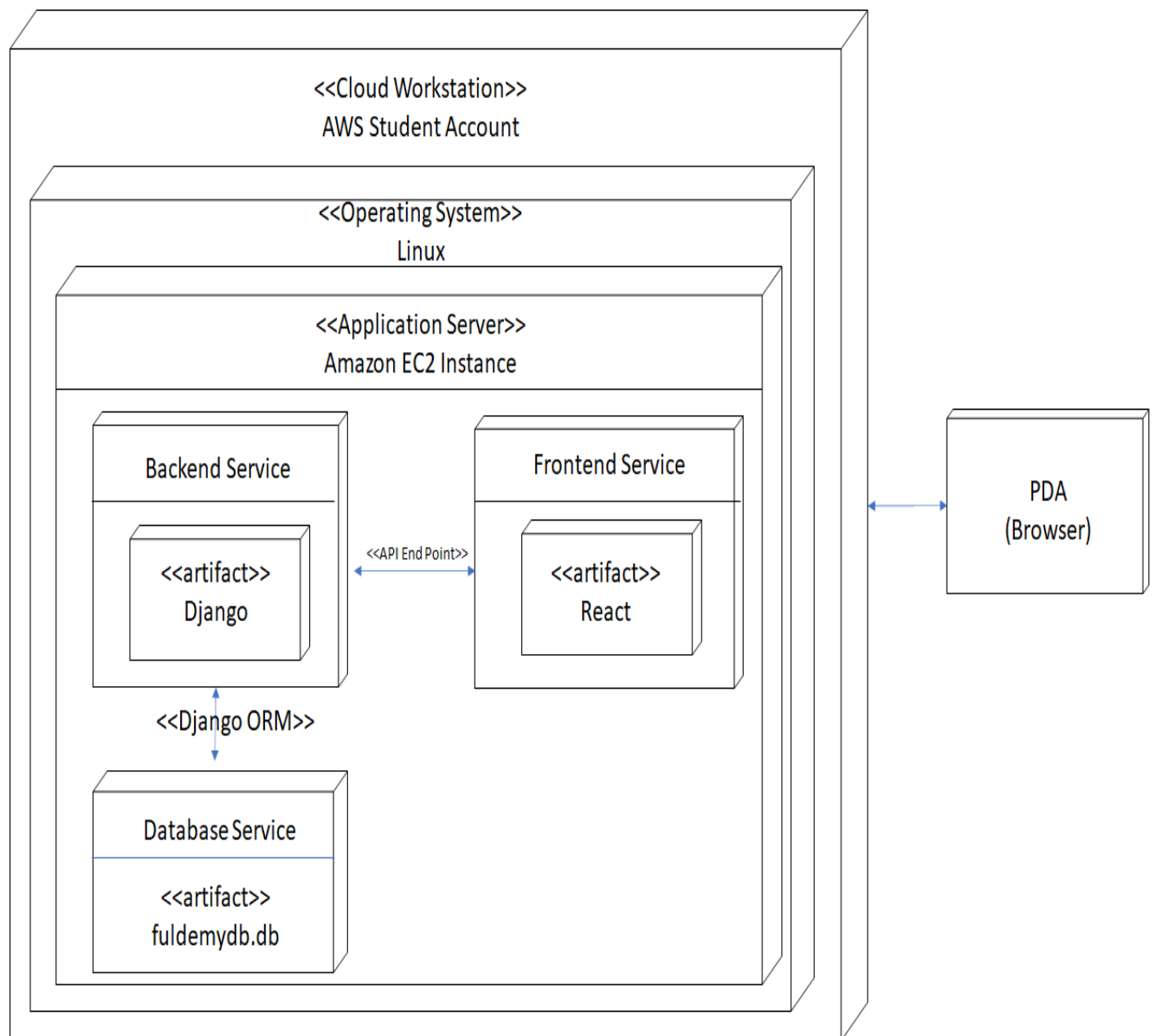


fig. Deployment Diagram

6. Key Risks

Schedule Risk

- Covid situation
 - ❖ Due to increasing incidences of covid, there is a chance of another lockdown. When that happens the weekly team meetings and the meetings with the CTO may need to be done online.
 - ❖ Some tasks may need to be completed together by being physically present. If the covid situation does not improve and the government mandates a lockdown, this will not be possible.
- Time offered by each team member towards the project
 - ❖ It is very important for every member to spend at least 6 hours on the project every week. Otherwise, some tasks might not be completed before the deadline.
- Personal time off without prior notice
 - ❖ During the course of the project, if one or more team members may have some need to travel to their home country or take some Personal time off without prior notice, it might cause the team spirit to fall and might end up causing the whole team to exceed deadlines.
- Illness or physical impairment
 - ❖ Sickness is an uncertain issue and can happen to anyone at any time especially in a pandemic situation over which nobody has control. In an unfortunate scenario if that happens to any one of the teammates, it might cause the team to exceed the deadline.
- Damage or loss of workstations
 - ❖ In an unfortunate situation of the impairment of laptops or desktops of team members, they may not be able to work for sometime, maybe even for weeks as most team members cannot afford backup devices.

Technical Risk

- Integration between front end and back end
 - ❖ SQL string is not properly written or gathered and UI works well but the database is not updated.
 - ❖ UI confirms that the user has successfully done something but the data transaction actually failed at some point.

Teamwork Risk

- Smooth integration of frontend and backend.
- Difference in culture and mentality.

- Team member shortage
 - ❖ If one or more team members take the decision to drop the course, it might be difficult for the rest of the team to fill the void. In this case, the team can request the CTO to reduce the scope of the project.

7. Project Management

As we described in the M1 document, Ms. Murugan is our Team lead. Mrs. Ray is the Backend Lead and Mr. Syed will work along with her as a backend developer. Mr. Dipesh is our Frontend Lead and Mr. Mujeeb will work with him as frontend developer and also Mr. Soyam will work as a QA Lead and Ms. Murugan will help him.

For the backend tasks we are planning to complete it by the Friday of every week and Mrs. Ray will ensure timely compilation and submission of backend tasks while for the frontend task Mr. Dipesh will be incharge. On Saturday Mr. Syed and Mr. Dipesh will integrate the backend and the frontend and fix any bugs. Mr. Soyam and Mr. Mujeeb will test this version on their respective local machines by Sunday noon and inform the development team of any issues. By Sunday evening Ms. Murugan will deploy the tested version in the cloud. On Monday morning Mr. Soyam and Ms. Murugan will do the final testing and even then, if any bugs appear then inform the development team and get it fixed at the earliest. By Tuesday afternoon Ms. Murugan will do the final deployment and send the mail to inform the CTO.

For the tracking purposes, we as a whole team decided that we are going to use Google Sheets. This is approved by the CTO.