

**Professor: Azer Karadag**

**COMP 246**



**TutorFlow**

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Part A: Project Scope and Requirements</b>	<b>5</b>
Section 1: Problem Statement	5
1.1.a. Problem & Need	5
1.1.b. List of Capabilities and Benefits	5
1.2 Identify the stakeholders and their roles	6
1.3 Identify sub-systems and functional components	6
1.4 Identify intended users of the SRS documentation	7
Section 2: General Overview Modelling	7
2.1 Context Flow Diagram (CFD)	7
Section 3: Requirements - Non-functional & Functional	7
3.1 Non- functional Requirements	7
3.2 Functional Requirements for three subsystems	8
3.2.1 Login and Registration Subsystems	8
3.2.2 Rating and Feedback Subsystems	12
3.2.3 Scheduling Subsystem	15
Section 4: Domain Class Diagram	18
4.1 List of classes	18
4.2 Diagram	19
Section 5: ERD	20
Section 6: UML System Sequence Diagrams	21
6.1 Login and Registration (Use Case: Login to the system)	21
6.2 Rating and Feedback (Use Case: Submit written feedback and ratings)	22

Section 7: State Diagrams	23
7.1 Authentication [for the User object]	23
7.2 Giving a rating [for the Student and Tutor objects]	24
Section 8: Technologies	25
8.1 Web Application	25
8.1.1 Client side (Front-end – GUI)	25
8.1.2 Business Logic (Middle layer - Class Methods):	25
8.1.3 Database	25
8.1.4 Additional Considerations	25
8.2 Mobile App	25
8.2.1 Client side (Front-end – GUI)	25
Section 9: Project Management (Gantt Chart)	26
<b>Part B: Software Design Architecture</b>	<b>27</b>
Section 1: Requirements Edits to Part A	27
1.1 Modifications to ERD and Class Diagrams (Section 4.2 and 5)	27
Section 2: Overview Model	27
2.1 Intended users of the SDD document	27
2.2 Context Flow diagram (CFD – What) versus Architectural Context Flow Diagram (ACFD – How)	28
2.2.1 A Context diagram (What overview)	28
2.2.2 Architectural Context diagram (How overview)	28
Section 3: Modularization	29
3.1 Partition the analysis model	29
3.1.1 Login Subsystem	29
3.1.2 Search Subsystem	30
3.1.3 Rating/Feedback Subsystem	31
3.2 Class Responsibility Collaboration (CRC)	32
3.2.1 Login Subsystem	32
3.2.2 Search Subsystem	33

TutorFlow	
3.2.3 Rating/Feedback Subsystem	34
3.3 TutorFlow Design Classes Diagram	35
3.3.1 Login Subsystem	35
3.3.2 Search Subsystem	36
3.3.3 Rating/Feedback Subsystem	37
Section 4: Framework M(odel) V(iew) C(ontroller)	38
4.1 MVC pattern diagram to include	38
4.1.1 Login Subsystem	38
4.1.2 Search Subsystem	39
4.1.3 Rating/Feedback Subsystem	40
4.2 Full Sequence Diagrams	41
4.2.1 Use case: Update Password (Login Subsystem)	41
4.2.2 Use case: Replying to feedback (Rating/Feedback Subsystem)	42
4.3 State Machine Diagrams	42
4.3.1 Use Case: Update Password [for User and Login Interface objects]	42
4.3.2 Use Case: Replying to feedback [for Tutor and Feedback objects]	43
Section 5: Data Layer	43
5.1 Database Schema	43
5.2 Technology List Update	44
Section 6: Gantt chart update	44
<b>Part C: System Design Documents</b>	<b>45</b>
Section 1: Corrections to Design Specifications Part B	45
Section 2: Software Design Patterns	45
2.1 Observer Pattern	45
2.2 Facade Pattern	45
2.3 Adapter Pattern	46
Section 3: Using common software design patterns	46
Section 4: UX/UI Design	48
4.1 Login Page	48

## TutorFlow

4.2 View Feedback and Ratings for a Tutor	50
Section 5: High level Component/Deployment Diagram	52
Section 6: Update the Gannt chart to include Part C Tasks	53
Section 7: Project Presentation	53

# Part A: Project Scope and Requirements

## Section 1: Problem Statement

### 1.1.a. Problem & Need

In the modern world where education is an extremely important aspect of numerous people's lives, many students are finding it difficult to keep up with the pace of learning in different educational environments. This can be due to individual needs being often overlooked in conventional educational systems in elementary, high school or post secondary institutions such as college or university. There are several difficulties and issues including, but not limited to, overcrowded classrooms, lack of personalized attention, learning disabilities or special needs, technological disparities and mental health.

How can these problems be alleviated? What are some possible solutions? Our system, TutorFlow, introduces an online tutoring platform to empower students by providing them with resources to learn at their own pace, with the help of highly qualified tutors. The idea is to give students a plethora of choices when it comes to fostering a learning environment personalized for them. This includes allowing them to pick tutors that align with their learning style, their personal budget, and other educational needs. The platform will provide students a learning environment that is accessible, affordable and high quality, alleviating their issues when it comes to learning.

### 1.1.b. List of Capabilities and Benefits

(i) Capabilities:

1. Register and store user information.
2. Allow the tutors to share their teaching experience and qualifications on the platform for students' reference. Also allow them to list which subjects they would want to tutor in and the prices for each session.
3. Register information about subjects that are available inside the platform.
4. Provide a search tool that will allow users to filter the subjects, prices, and ratings to students and tutors.
5. Incorporate a secure payment gateway for transactions by debit card, credit card, and Apple pay.
6. Provide a scheduling system for students to book tutoring sessions with available tutors in respective subjects.
7. Allow for users to video call for their tutoring sessions or watch a pre-recorded video taken by the tutor.

8. Provide a built-in communication area between students and tutors. This will include both text-based and video-based communication.
9. Allow students to write text feedback on tutors they had sessions with. This will also include a rating system.
10. Allow parents of elementary or high school students to view their child's student progress report.

(ii) Benefits:

1. Provide students who are having difficulty in certain subjects or lessons with a personalized education system, enhancing their learning experience.
2. Empower tutors or teachers who want to earn extra money.
3. Develop tutors who want to gain teaching experience.
4. Provide a safe and secure platform where payments are handled with utmost security.
5. Promote a collaborative community for learners and teachers alike, that allows for easy knowledge sharing or networking.
6. Encourage continuous improvement for tutors as students are able to write real-time ratings and feedback, helping them improve their teaching methods.

## 1.2 Identify the stakeholders and their roles

- Students
- Tutors
- Parents of students concerned about the educational progress of their children.
- Technical development team working to maintain and improve the platform's technical infrastructure. They will also be referred to as **admin**.

## 1.3 Identify sub-systems and functional components

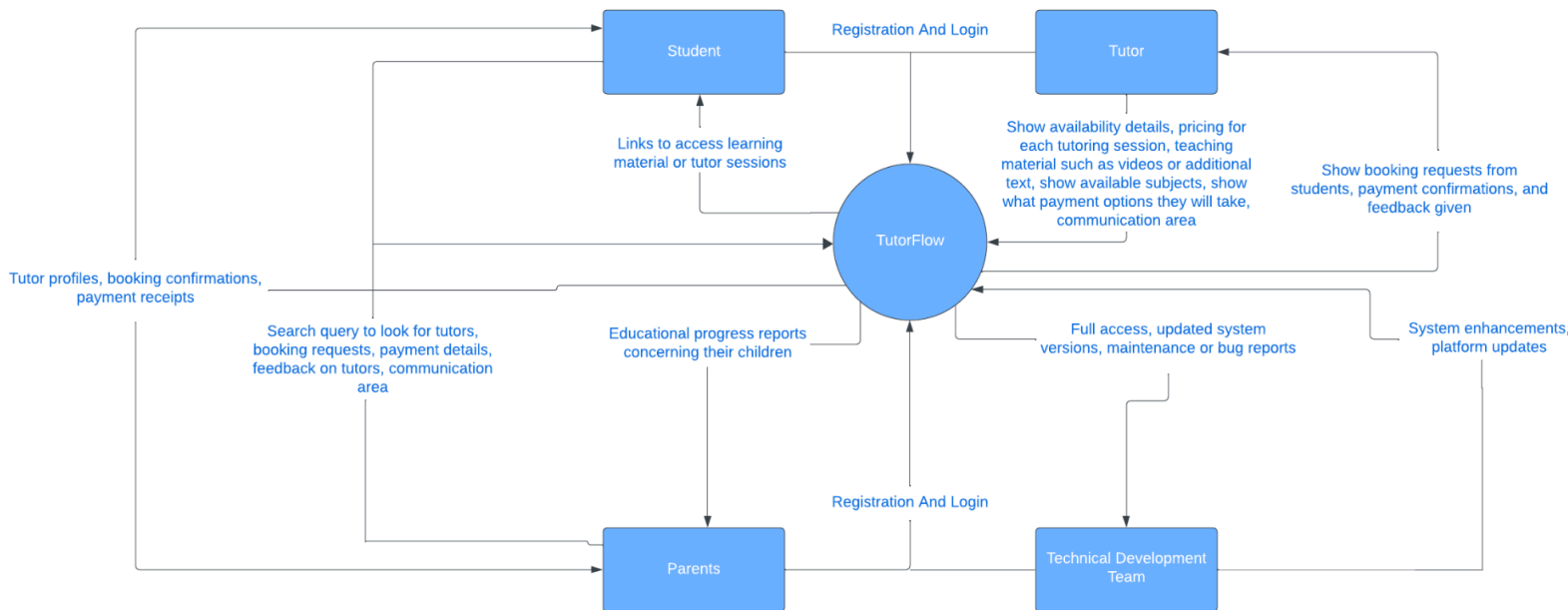
- Log In subsystem
- Registration subsystem
- Search Subsystem
- Payment Subsystem
- Profile Management Subsystem
- Rating/Feedback Subsystem
- User Communication Subsystem
- Video Library Subsystem
- Scheduling Subsystem

## 1.4 Identify intended users of the SRS documentation

- Software Design Architects
- Software Developer
- Tester/QA
- Database Administrator

## Section 2: General Overview Modelling

### 2.1 Context Flow Diagram (CFD)



## Section 3: Requirements - Non-functional & Functional

*\* Note that for this Software Design project the emphasis is placed on Functional Requirements \**

### 3.1 Non- functional Requirements

NFR#	Name	Description
NFR001	Efficiency	The system should ensure quick response times and be able to handle a high number of simultaneous users



		without reduced performance.
NFR002	Scalability	The platform should be designed to handle an increase in the user base and data volume.
NFR003	Security	The system must protect sensitive data including personal information and payment details from unauthorized access.
NFR004	Maintainability	The system should be structured in a way that facilitates easy updates and maintenance, allowing for quick resolution of issues.
NFR005	Accessibility	The system should comply with the latest accessibility standards to assist users with disabilities.
NFR006	Flexibility	The platform should adapt to different user preferences and requirements, offering a personalized user experience based on individual learning and teaching styles.
NFR007	Interoperability	The system should easily integrate with other tools and platforms, such as payment gateways or educational resources, providing a cohesive user experience.

## 3.2 Functional Requirements for three subsystems

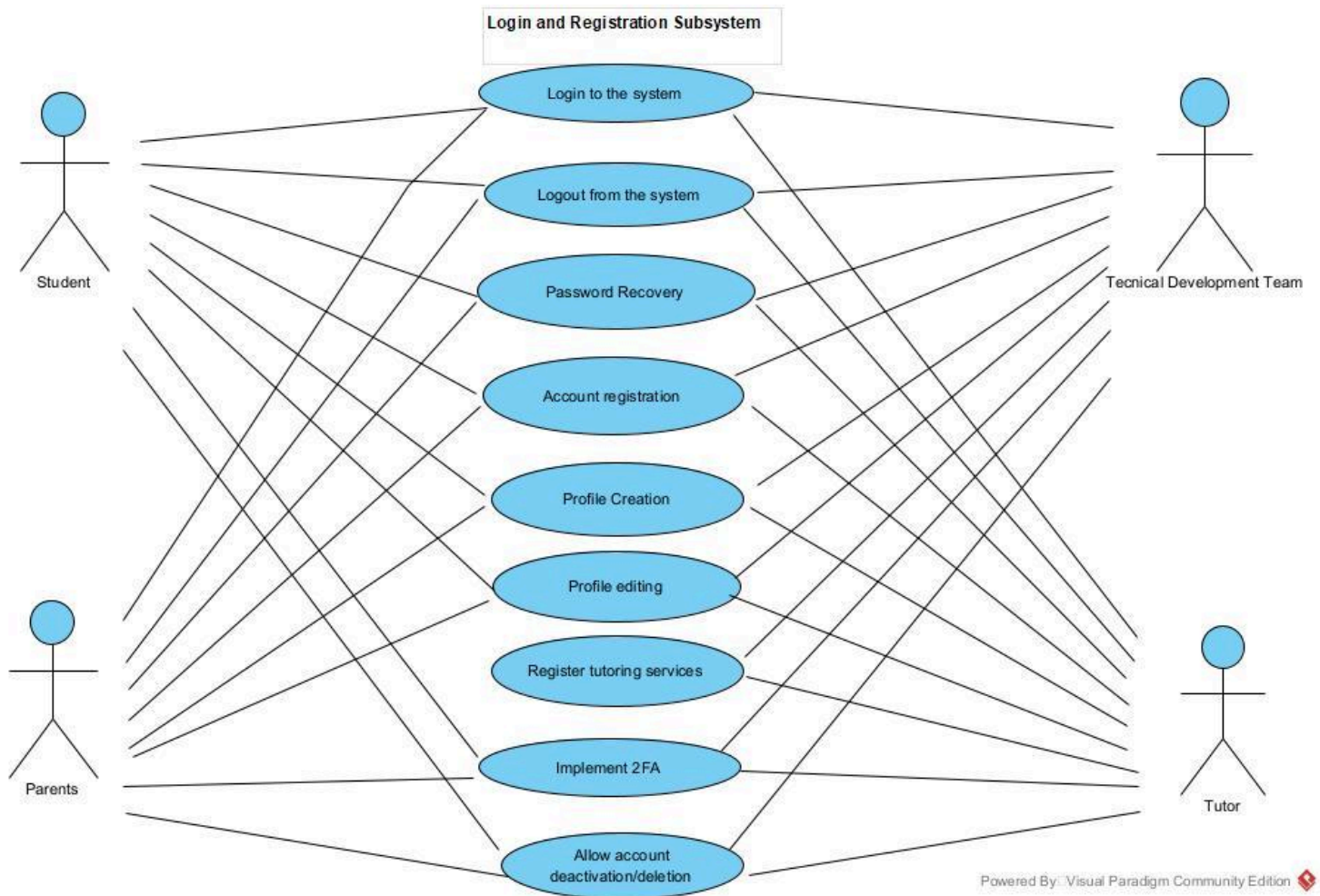
### 3.2.1 Login and Registration Subsystems

FR#	Name (Goal Use case)	Role Player	Description
FR01	Login to the system	All users	Allow users to log into the system to access their personal accounts using a secure password and username/email.
FR02	Logout from the system	All users	Enable users to log out from the system at any time.
FR03	Password recovery	All users	Provide users with the ability to recover their password through a secure method, such as sending a recovery link to their registered email.

## TutorFlow

FR04	Account registration	All users	Allow new users to register for an account by providing necessary details such as email, username, and password.
FR05	Profile creation	All users	After registering, users should be able to create a profile, add personal details, and set up preferences.
FR06	Profile editing	All users	Users should be able to edit their profile information, including updating their profile picture, changing their preferred name, etc.
FR07	Register tutoring services	Tutors, Admin	Allow tutors to register their services and choose the subjects they are teaching.
FR08	Implement 2FA	All users	Implement two-factor authentication where users verify their identity through another method besides the password, such as a mobile app, SMS, or a passcode number sent by email.
FR09	Allow account deactivation/deletion	All users	Provide users with the option to permanently delete their account, removing all personal data from the system.

## TutorFlow



### User Story

1. As a user I want to login or register to create my account, so I can access the system.

#### Acceptance Criteria:

- Should be able to create a profile
- Should be able to register with a username and a password.
- Should be able to verify if the user is already registered.
- Should be able to see the users last login date time.

2. As a user I want to recover my forgotten password, so I can regain access into the system.

#### Acceptance Criteria:

- Should be able to initiate a password recovery process.

## TutorFlow

- Should receive a recovery link in the registered email.
- Should be able to change the password.
- Should be able to get advice when the user's password expires.

3. As a user, I want to have a two-factor authentication option, to have an extra layer of security for my account.

Acceptance Criteria:

- Should be able to set up two-factor authentication using various methods like mobile apps or SMS.
- Should receive a verification code during the login process when two-factor authentication is enabled.

4. As a user, I want to edit my profile information, so that I can keep my profile updated with the latest details.

Acceptance Criteria:

- Should be able to change personal details such as location, profile pictures, and personal information such as address or phone number
- Should be able to change the type of user they are, either student or tutor.

5. As a user, I want to logout of the system at any time.

Acceptance Criteria:

- Should be able to logout to stop access to the system with ease.

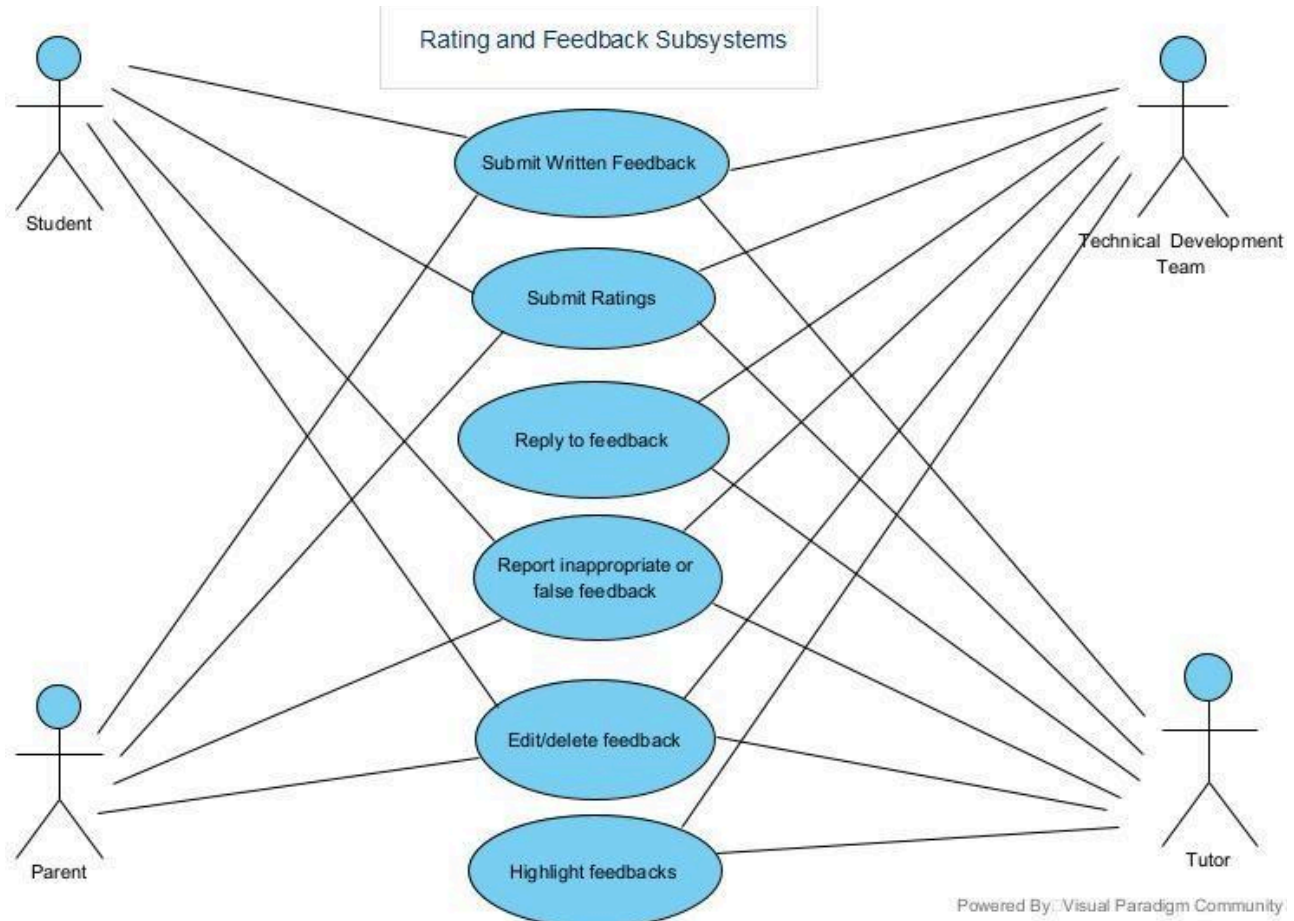
6. As a user, I want to permanently delete my account when I no longer wish to use the service.

Acceptance Criteria:

- Should be able to initiate the account deletion process from the account settings.
- Should receive a confirmation prompt to prevent accidental deletions.

## 3.2.2 Rating and Feedback Subsystems

FR#	Name (Goal Use case)	Role Player	Description
FR01	Submit written feedback	All users	Users can submit their feedback for their experience in tutorials or teaching materials for evaluations.
FR02	Submit ratings	All users	Users can express their level of satisfaction based on their experience in sessions.
FR03	Reply to feedback	Tutors, Admin	Tutors are able to reply to students' feedback.
FR04	Report inappropriate or false feedback	All users	Users are allowed to report unreal or misleading feedback with proof, e.g. tutorial recording, teaching materials, and etc..
FR05	Edit/delete feedback	All users	Users are allowed to update/delete their feedback
FR06	Highlight feedbacks	Tutors, Admin	Allow tutors to highlight feedback from students and parents in their profile for other users' review.



## User Story

1. As a user, I want to submit a written feedback .

Acceptance criteria:

- Users must be able to easily find and navigate to past sessions.
- Users can enter and submit feedback in a designated input area.
- Upon submission, a confirmation message should be displayed.
- The submitted feedback should be visible and stored properly in the system.

2. As a user, I want to submit ratings.

Acceptance criteria:

- Users can select a rating between 1 to 10, where the scale and meaning (1 being lowest and 10 being highest) is clearly defined.
- A clearly labeled button or option should be available for submitting ratings.
- Upon submission, a confirmation message should be displayed.

- The system accurately updates and displays the new average rating for the student/tutor.

3. As a user, I want to reply to feedback.

Acceptance criteria:

- A reply option should be available and clearly marked for each piece of feedback.
- Replies, once submitted, should be visible to the respective student/parent and stored properly.
- Users should be notified when their feedback receives a reply.

4. As a user, I want to report inappropriate or false feedback.

Acceptance criteria:

- An option to report feedback is clearly visible and accessible.
- Users can submit their report with optional additional comments or evidence.
- A confirmation message is displayed upon submission of a report.
- Reported feedback is flagged for review by admins.

5. As a user, I want to edit/delete feedback.

Acceptance criteria:

- Users can find and select their previously submitted feedback.
- Clear options for editing and deleting feedback are available.
- Changes to feedback (edits or deletions) are saved and reflected immediately.
- A confirmation message is displayed for any change to feedback.

6. As a user, I want to highlight feedback.

Acceptance criteria:

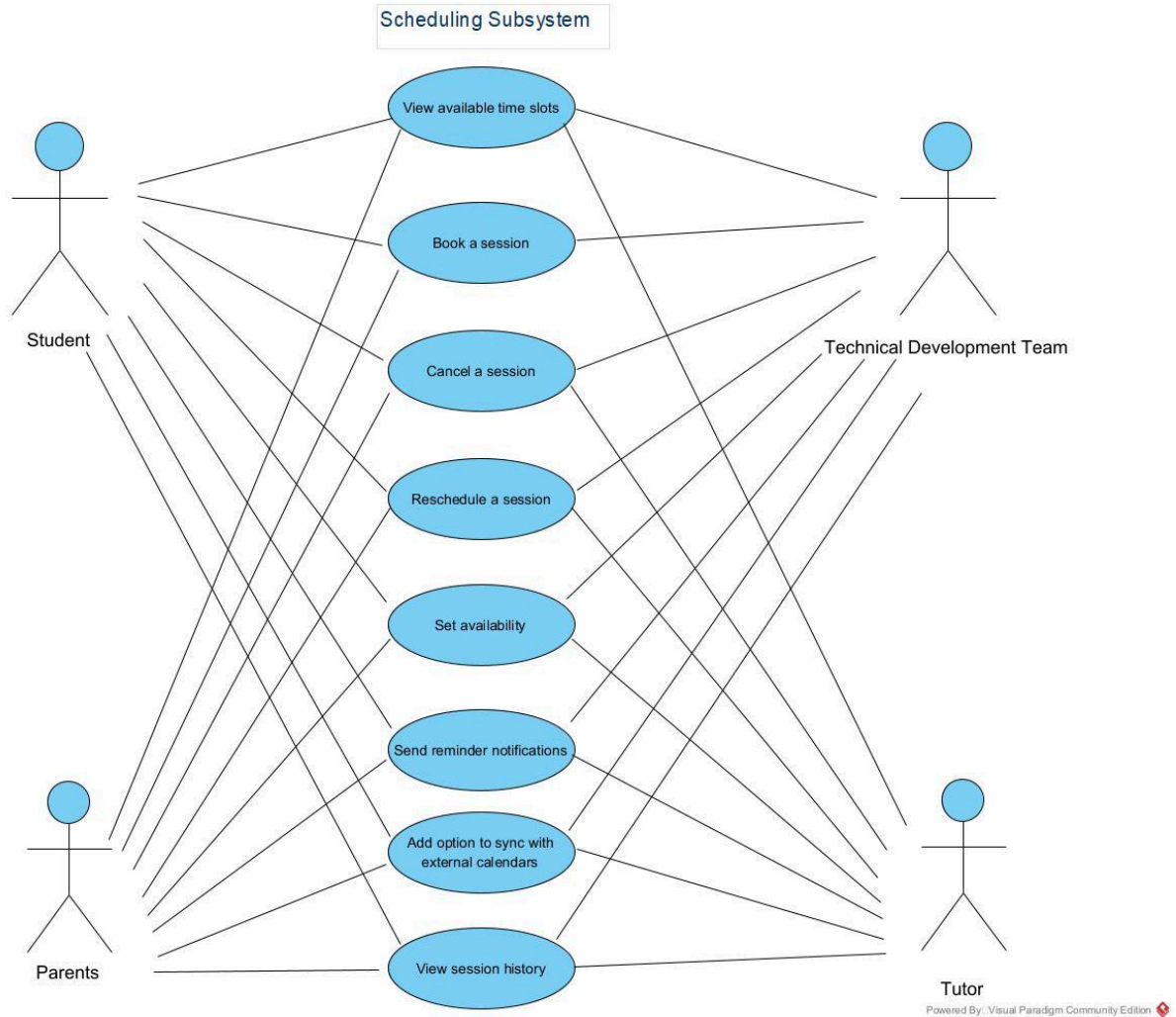
- Tutors can easily navigate to and view feedback received.
- An option to "Highlight" feedback on their profile is visible and easy to use.
- Highlighted feedback is prominently displayed on the tutor's profile.
- Tutors can easily un-highlight feedback if desired.

## 3.2.3 Scheduling Subsystem

FR#	Name (Goal Use case)	Role Player	Description
FR01	View available time slots	All users	The system should integrate the schedule of tutors and students to present the available time slots for them to choose tutorial sessions.
FR02	Book a session	Students, Parents, Admin	Students can book tutorial sessions based on available time slots.
FR03	Cancel a session	All users	After session confirmation, users can apply for session cancellation with explanations, and the system would notify the counter party for their acceptance.
FR04	Reschedule a session	All users	Users can reschedule their sessions based on available time slots, and the system would notify the counter party.
FR05	Set availability	All users	Users can update their available time slots, and the system will warn them of the possible changes on tutorial sessions required after that change.
FR06	Send reminder notifications	All users	Users can send reminder notifications to each other in relation to session start time, tutorial exercise, etc.
FR07	Add option to sync with external calendars	All users	Users can synchronize their schedule of the platform with their external calendars to avoid time conflicts.
FR08	View session history	All users	Users can review the session history of each other, such as number of session reschedulings and session cancellations..



# TutorFlow



## User Story

1. As a user, I want to view available time slots of tutors and students to plan and schedule tutorial sessions.

Acceptance criteria:

- Available time slots should be visibly marked or indicated in a scheduling interface.
- The schedule should be in a readable and understandable format (e.g., a calendar view).

2. As a student or parent, I want to book tutorial sessions based on the available time slots to ensure organized learning.

Acceptance criteria:

- Available time slots should be visibly marked or indicated in a scheduling interface.
- The schedule should be in a readable and understandable format (e.g., a calendar view).

3. As a user, I want to cancel a booked session by providing an explanation, and notify the other party of the cancellation.

Acceptance criteria:

- Users can find and select their booked sessions easily.
- An option to cancel and provide explanations is accessible and user-friendly.
- The other party is notified of the cancellation and provided with the explanation.

4. As a user, I want to reschedule sessions, notifying the other party of the change to ensure aligned schedules.

Acceptance criteria:

- Users can find and select their booked sessions easily.
- An option to cancel and provide explanations is accessible and user-friendly.
- The other party is notified of the cancellation and provided with the explanation.

5. As a user, I want to update my availability in the system to manage session bookings effectively.

Acceptance criteria:

- The system allows users to define and update their available time slots.
- Warnings regarding potential disruptions or rescheduling of existing sessions are displayed when availability is changed.

6. As a user, I want to send reminder notifications related to session timings, exercises, etc., to enhance communication and timeliness.

Acceptance criteria:

- Users can send reminders regarding upcoming sessions or relevant information.

- Reminders are received by the intended user and are visibly notified.
7. As a user, I want to synchronize my platform schedule with external calendars to manage all my commitments centrally.

Acceptance criteria:

- The system provides an option to synchronize with popular external calendar services such as Google Calendar or Apple Calendar.
  - Any additions or changes in the platform schedule are reflected in the external calendar and vice versa.
8. As a user, I want to view the session history, including rescheduling and cancellations, to keep track of past interactions and commitments.

Acceptance criteria:

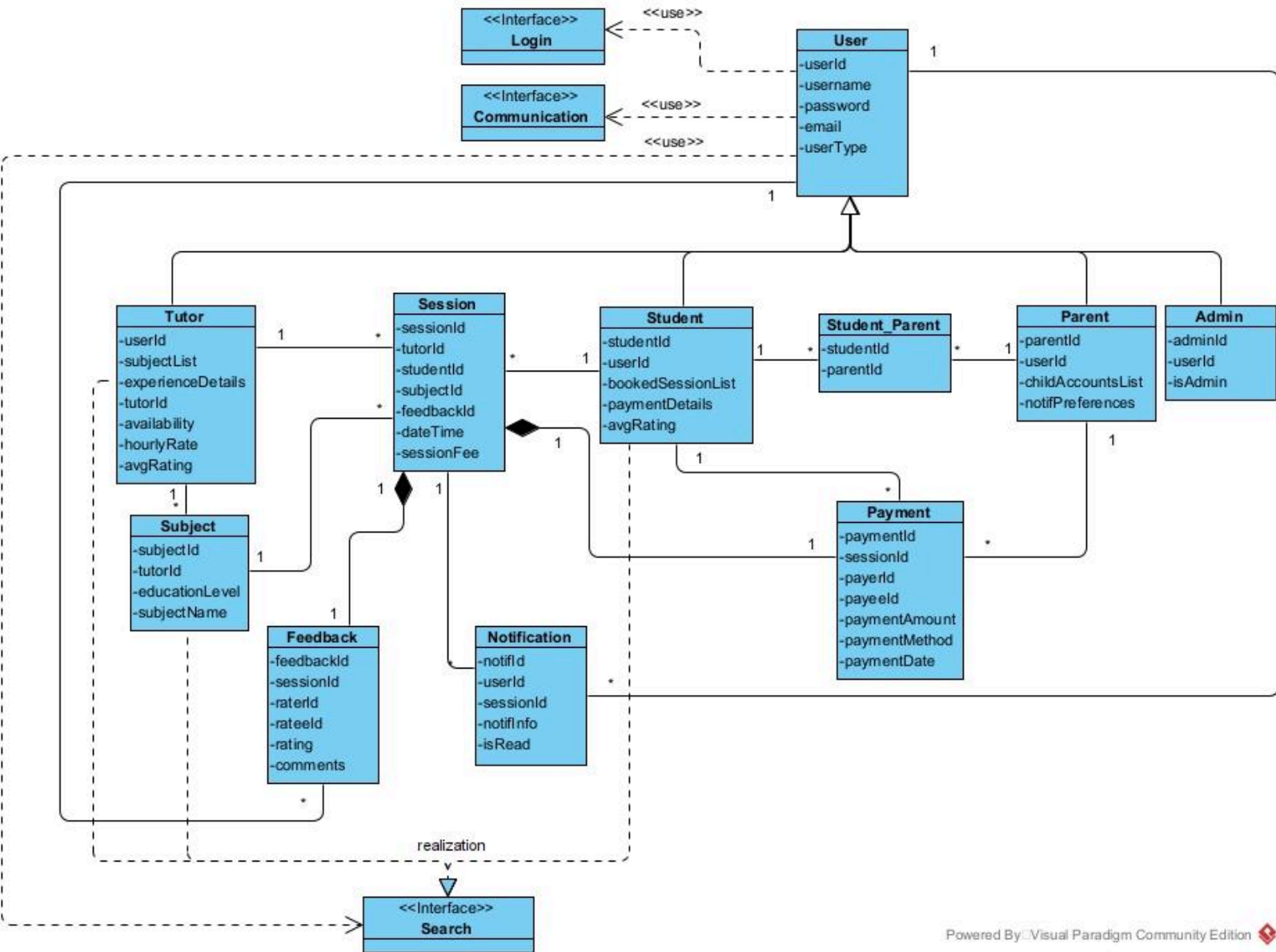
- Users can easily navigate to and view their session history.
- Session history includes details of initial schedules, rescheduling, cancellations, and relevant explanations.

## Section 4: Domain Class Diagram

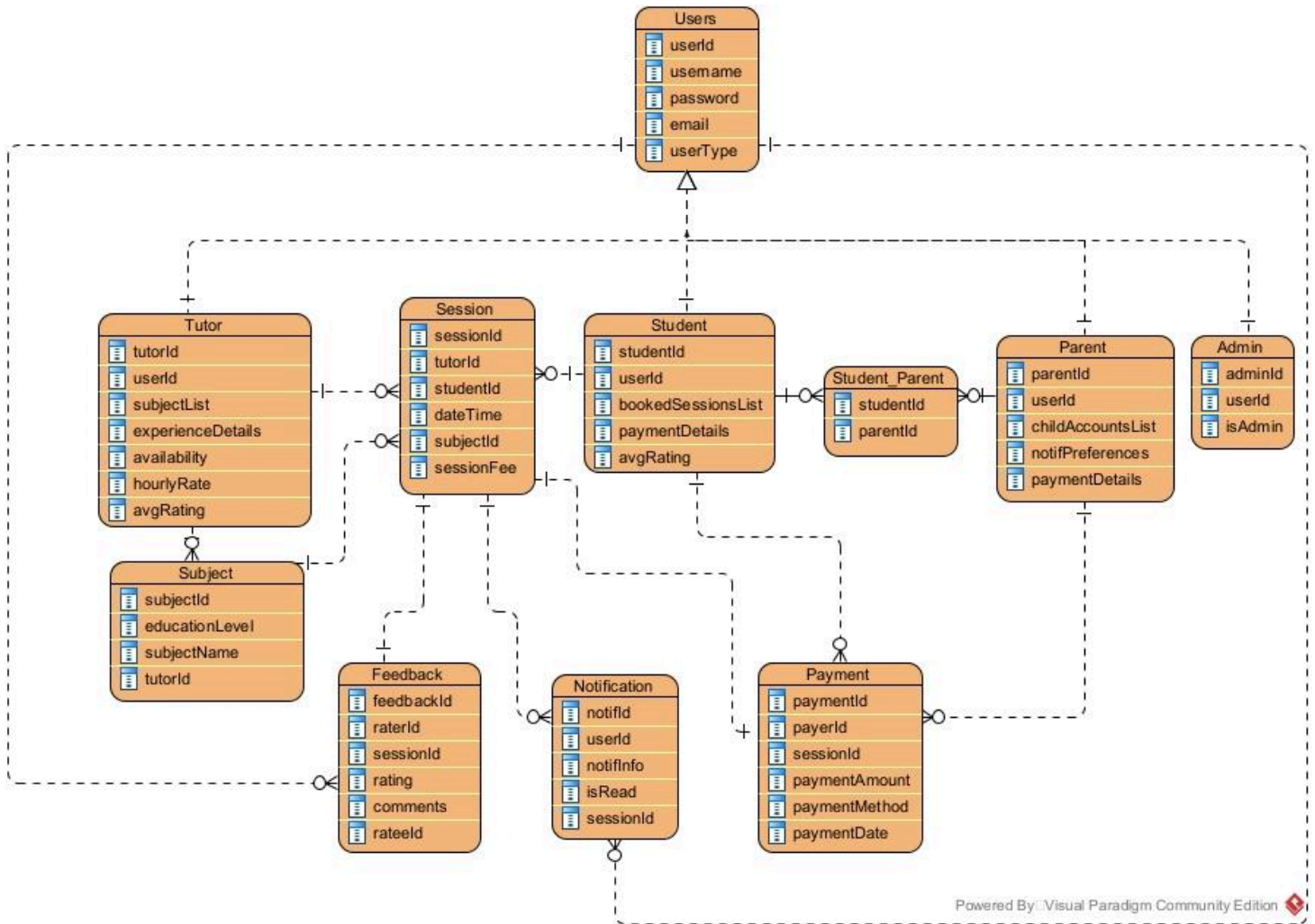
### 4.1 List of classes

- |                  |                |
|------------------|----------------|
| • User           | • Parent       |
| • Tutor          | • Subject      |
| • Student        | • Feedback     |
| • Session        | • Notification |
| • Payment        | • Admin        |
| • Student_Parent |                |

## 4.2 Diagram

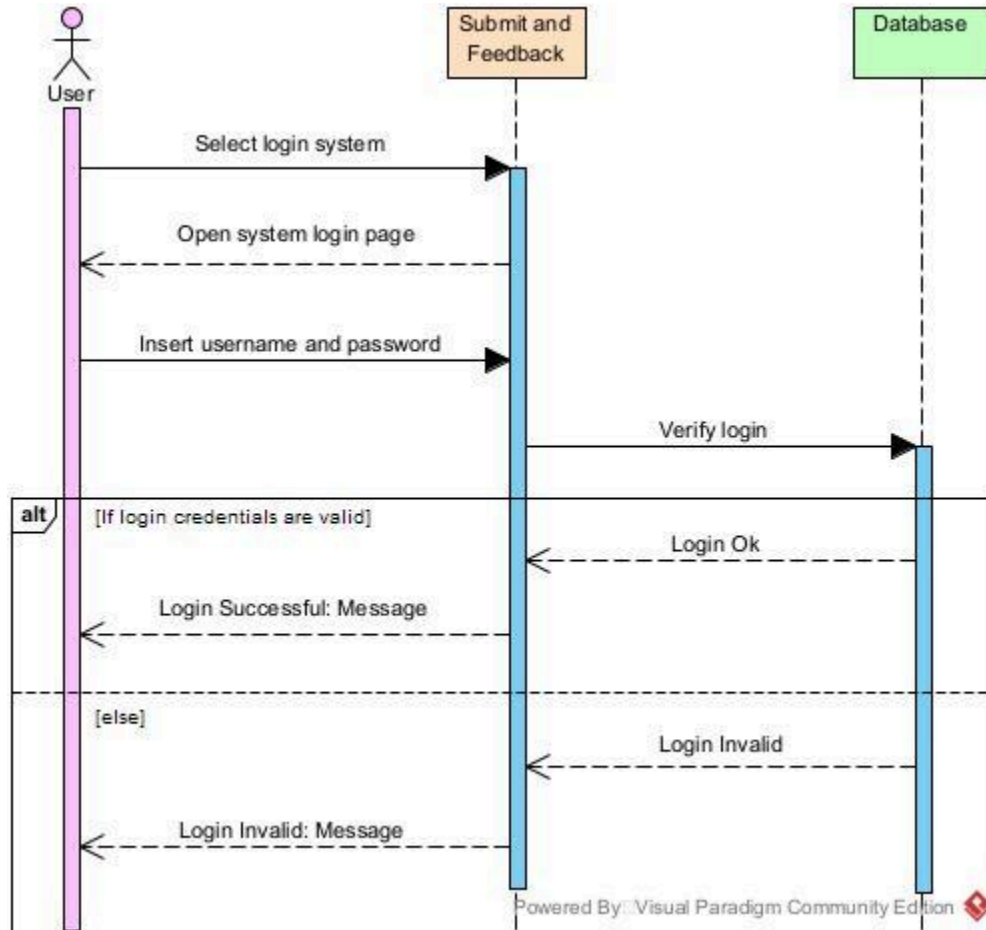


## Section 5: ERD

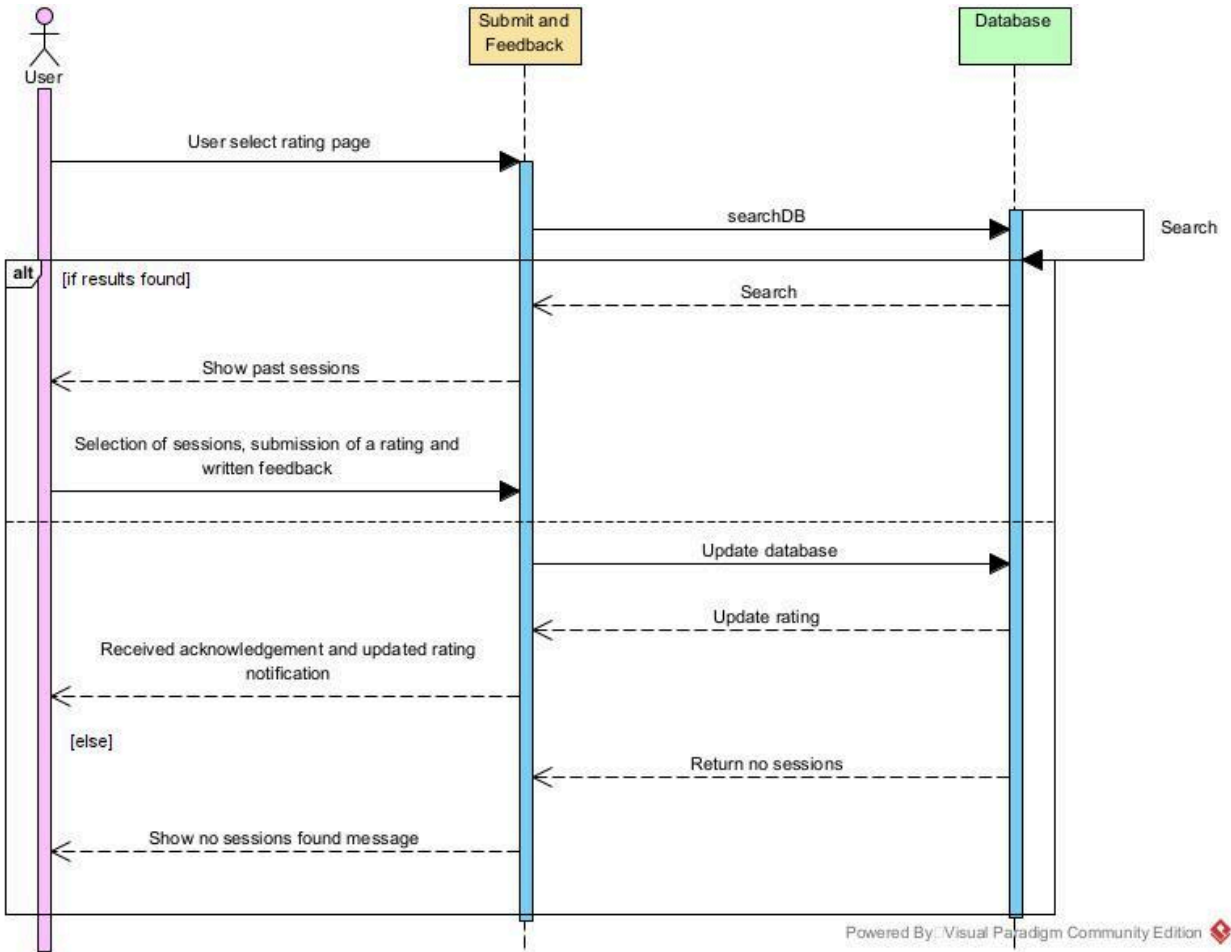


## Section 6: UML System Sequence Diagrams

### 6.1 Login and Registration (Use Case: Login to the system)

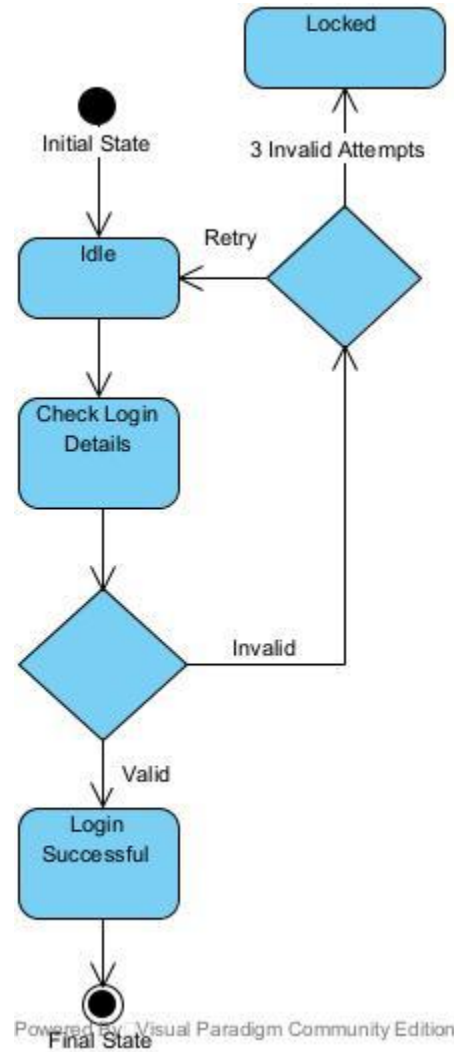


## 6.2 Rating and Feedback (Use Case: Submit written feedback and ratings)



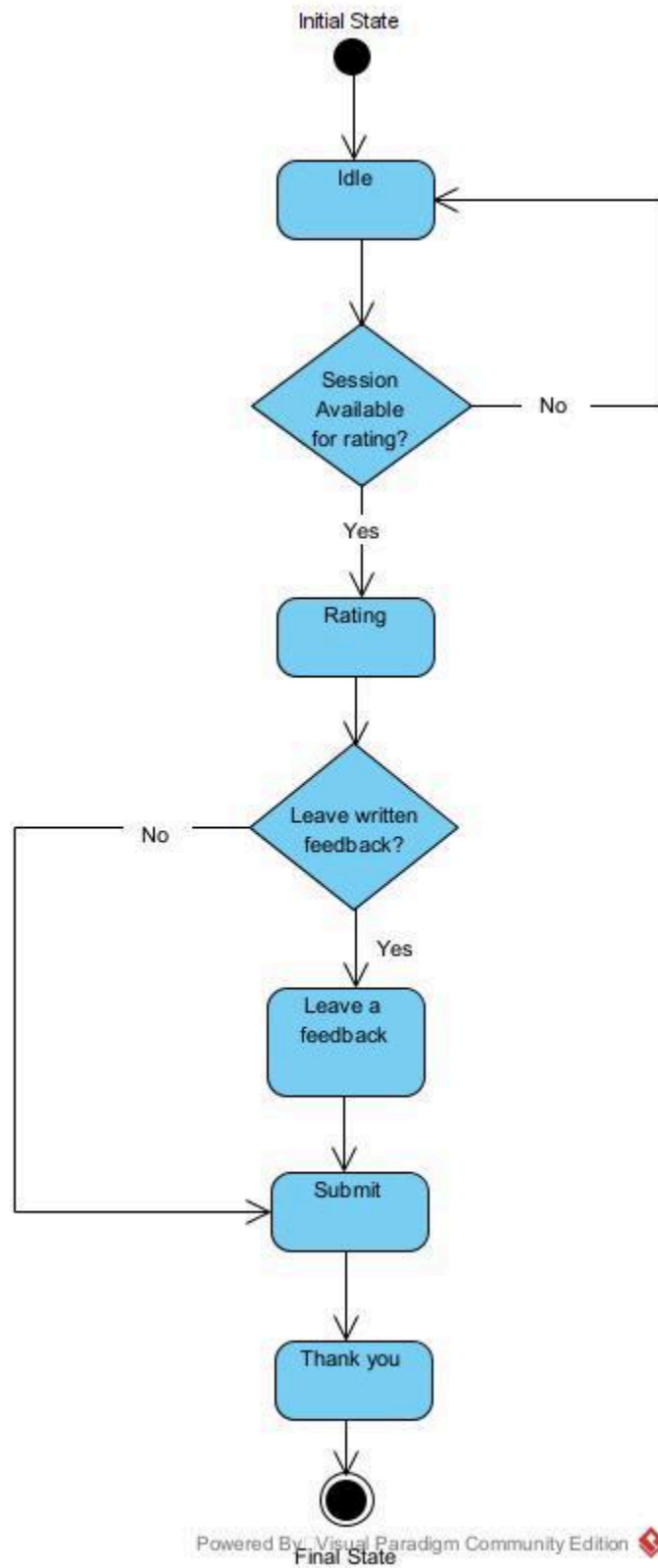
## Section 7: State Diagrams

### 7.1 Authentication [for the User object]





## 7.2 Giving a rating [for the Student and Tutor objects]



## Section 8: Technologies

### 8.1 Web Application

#### 8.1.1 Client side (Front-end – GUI)

- HTML, CSS, and JavaScript
- React.js
- Bootstrap

#### 8.1.2 Business Logic (Middle layer - Class Methods):

- Node.js
- Express.js
- Stripe API
- Mongoose
- Passport.js

#### 8.1.3 Database

- MongoDB

#### 8.1.4 Additional Considerations

- Docker - For containerization
- AWS - Hosting
- Gitlab CI - Continuous Integration/Deployment (CI/CD) pipeline
- Cypress - Automated testing

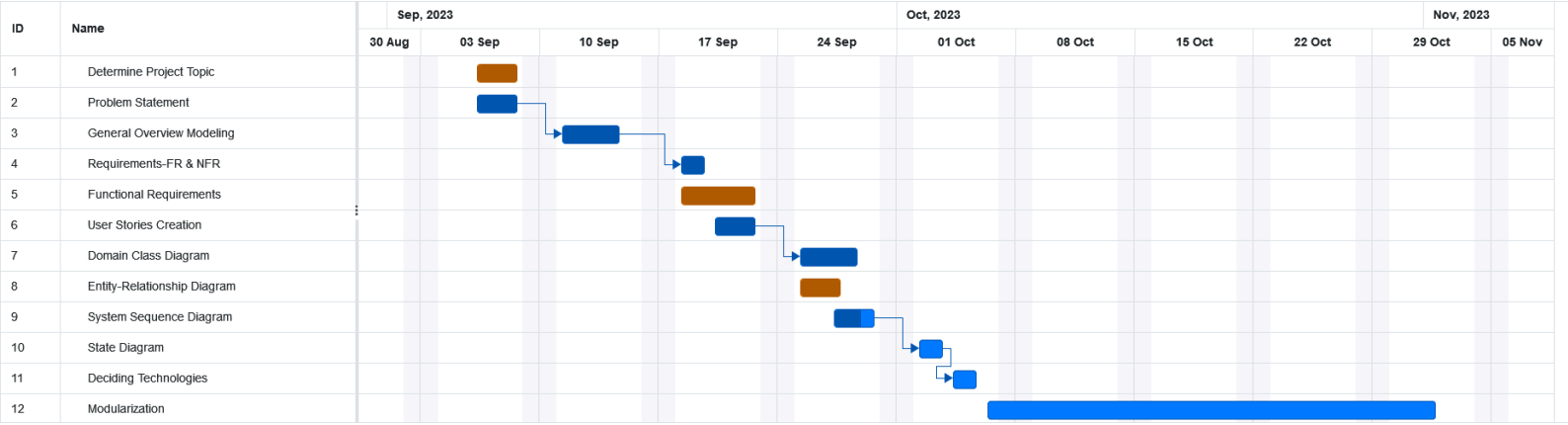
### 8.2 Mobile App

#### 8.2.1 Client side (Front-end – GUI)

- React Native
- Flutter
- Swift (iOS) and Kotlin (Android)

The technologies used for the business logic, database, and other additional considerations will be the same as the ones used for the web application version.

Section 9: Project Management (Gantt Chart)



## **Part B: Software Design Architecture**

### **Section 1: Requirements Edits to Part A**

#### **1.1 Modifications to ERD and Class Diagrams (Section 4.2 and 5)**

- Added primary keys to User subclasses in the ERD and class diagrams. They were previously missing primary keys and were only composed of foreign keys.

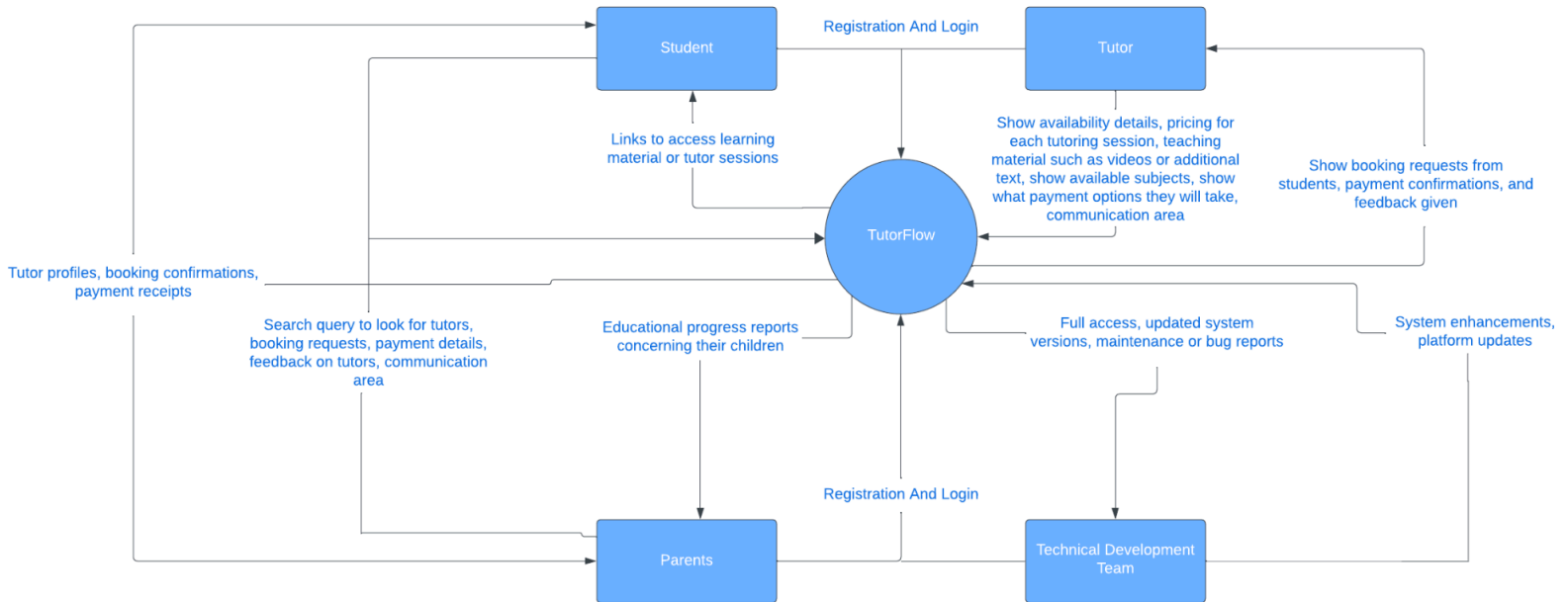
### **Section 2: Overview Model**

#### **2.1 Intended users of the SDD document**

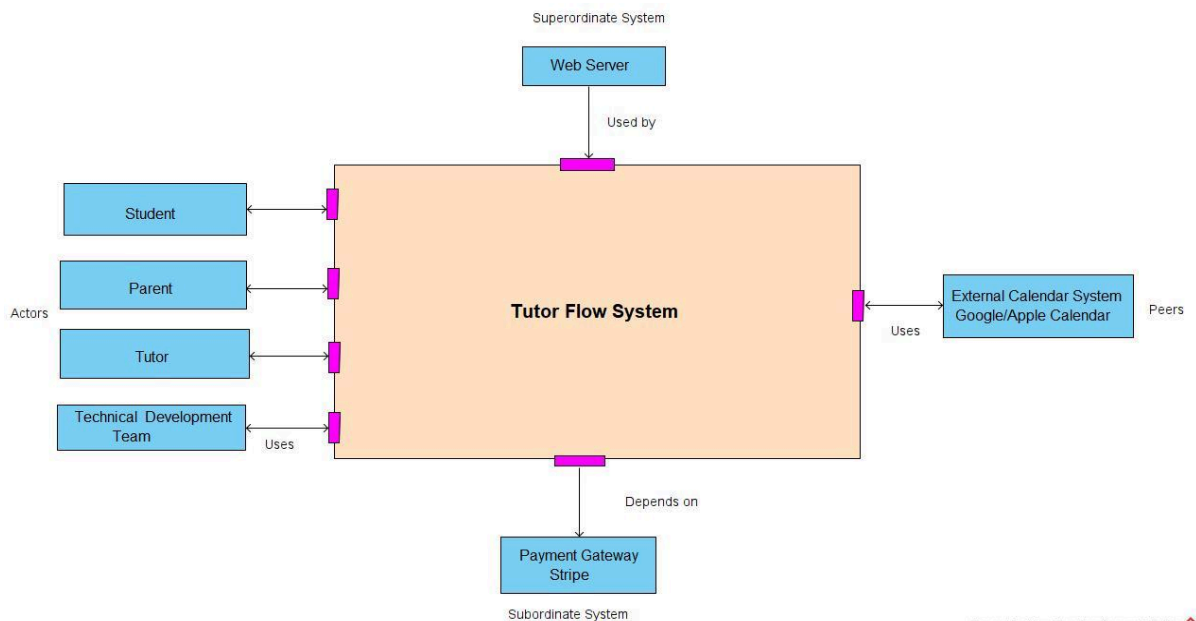
The purpose of the System Design Document (SDD) is to specify in detail the architecture, components and design choices for the system, ensuring a clear and comprehensive guide for its implementation. Therefore, the primary audience for this document are software developers tasked with bringing the design to life and the technical managers overseeing the project's progress and quality. Additionally, QA testers and system designers Together, they are responsible for the system's development and refinement.

## 2.2 Context Flow diagram (CFD – What) versus Architectural Context Flow Diagram (ACFD – How)

### 2.2.1 A Context diagram (What overview)



### 2.2.2 Architectural Context diagram (How overview)

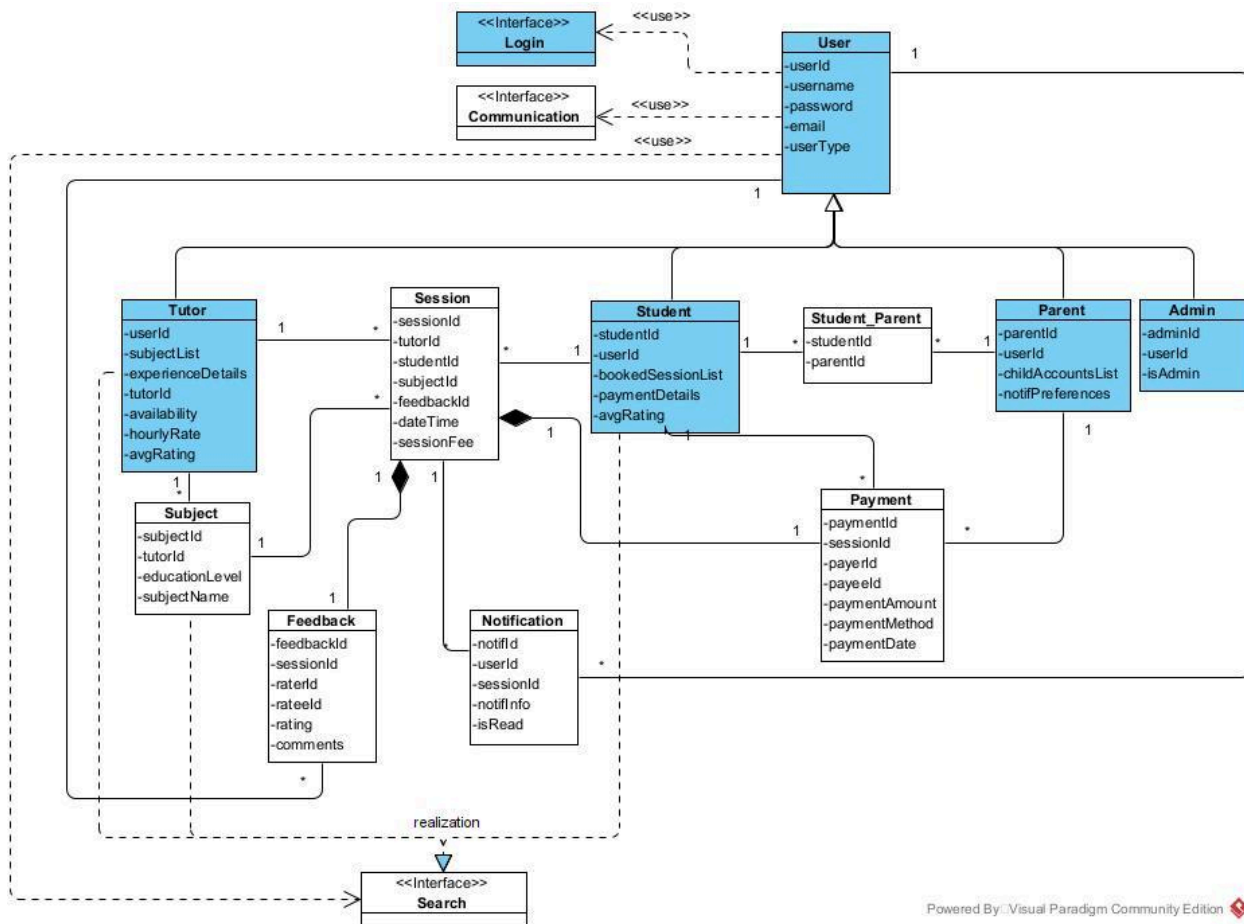


## Section 3: Modularization

### 3.1 Partition the analysis model

#### 3.1.1 Login Subsystem

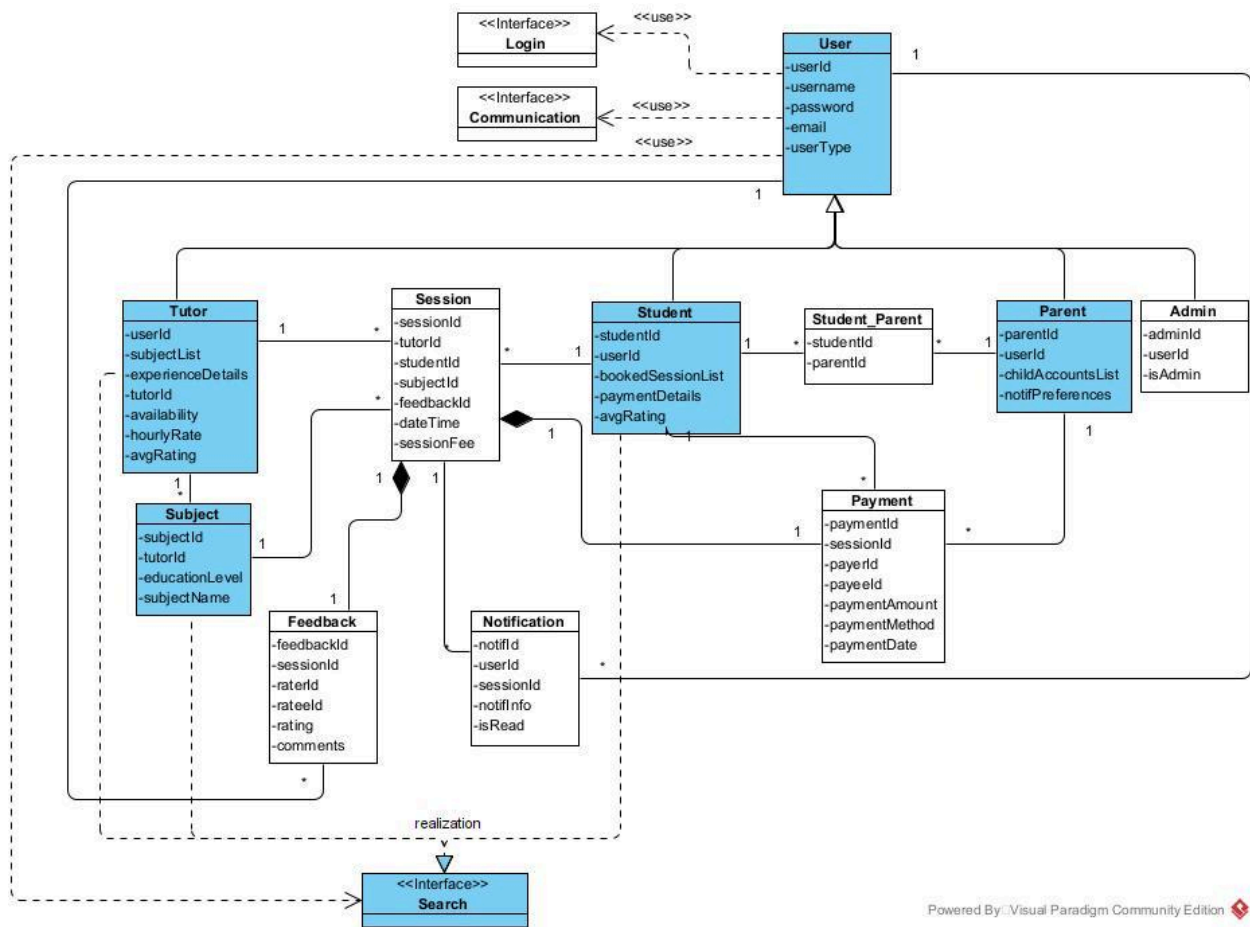
- Login Interface
- User
- Tutor
- Student
- Parent
- Admin



# TutorFlow

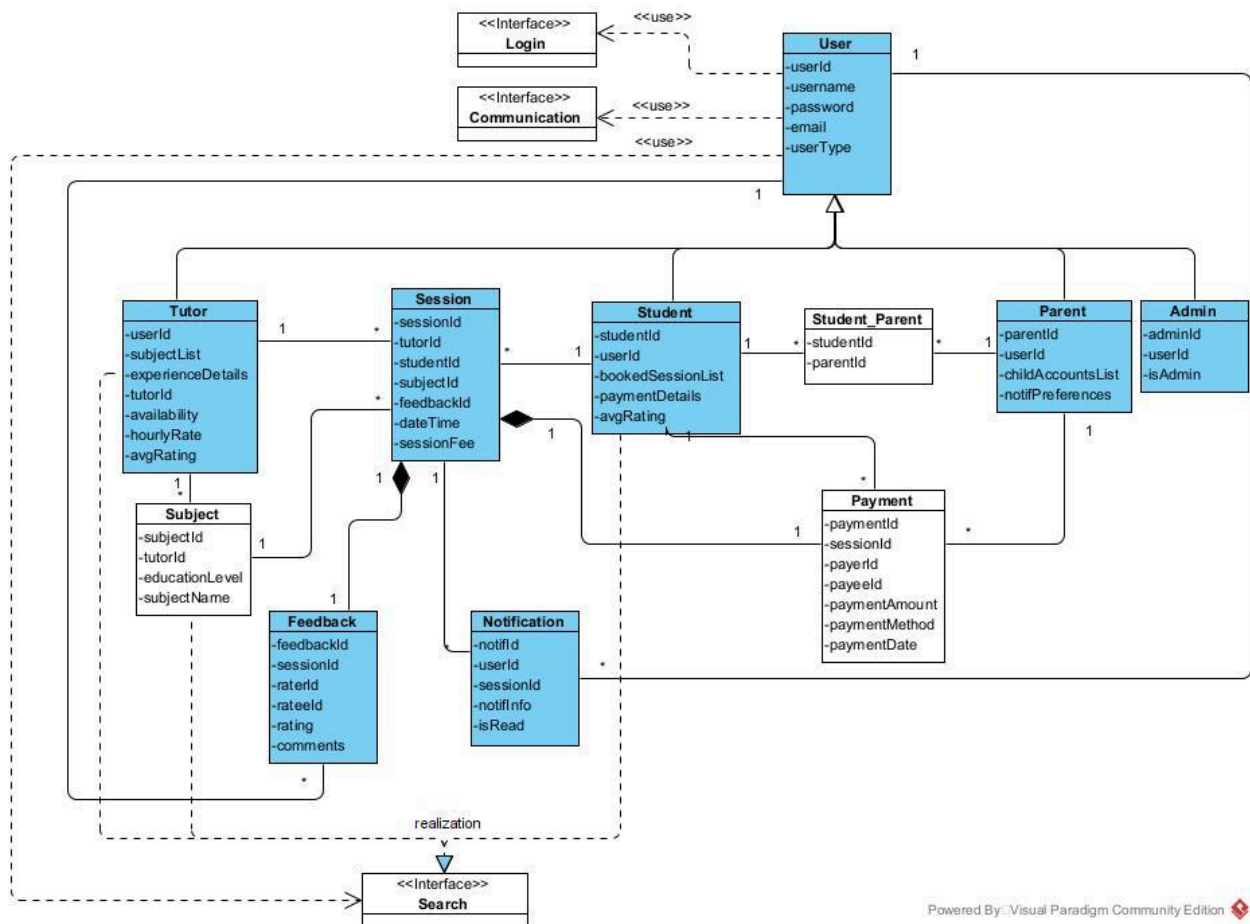
## 3.1.2 Search Subsystem

- Search Interface
- User
- Tutor
- Student
- Subject
- Parent



### 3.1.3 Rating/Feedback Subsystem

- User
- Tutor
- Student
- Parent
- Rating
- Feedback
- Session
- Admin
- Notification





## 3.2 Class Responsibility Collaboration (CRC)

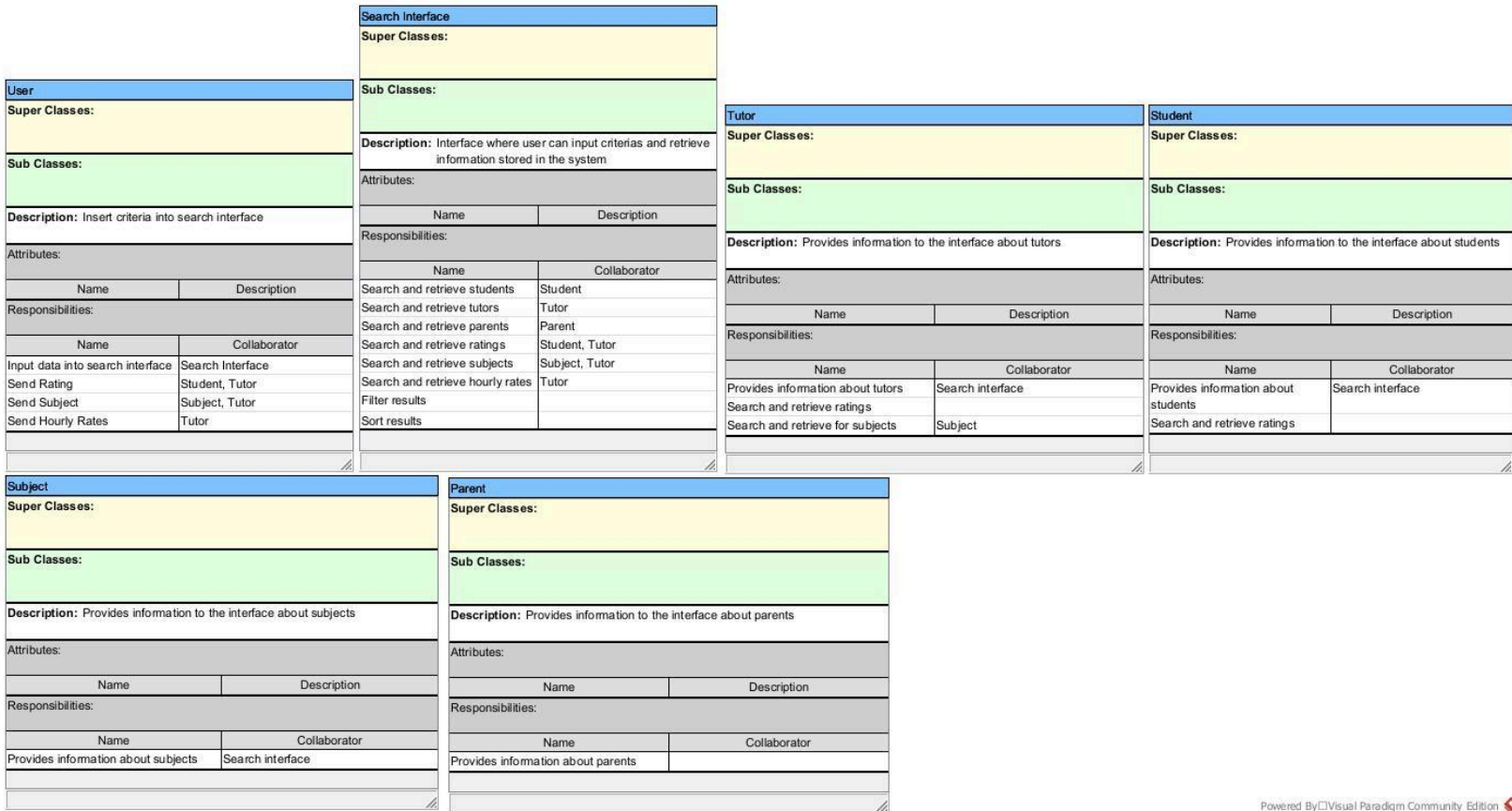
### 3.2.1 Login Subsystem

<div>User</div> <div>Super Classes:</div> <div>Sub Classes: Student, Parent, Admin, Tutor</div> <div>Description: Identify the user and store their information</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Provide username</td><td>Login interface, Student, Parent, Admin,</td></tr><tr><td>Provide password</td><td>Login interface</td></tr><tr><td>Allow user to change password</td><td>Login interface</td></tr><tr><td>Logout</td><td>Login interface</td></tr></table>		Name	Description	Name	Collaborator	Provide username	Login interface, Student, Parent, Admin,	Provide password	Login interface	Allow user to change password	Login interface	Logout	Login interface	<div>Login interface</div> <div>Super Classes:</div> <div>Sub Classes:</div> <div>Description: Checks if user can login, requests for system privileges and user information</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Checks username and password</td><td>User</td></tr><tr><td>Request for system privileges</td><td>User, Student, Parent, Admin, Tutor</td></tr><tr><td>Request for user information</td><td>User, Student, Parent, Admin, Tutor</td></tr></table>		Name	Description	Name	Collaborator	Checks username and password	User	Request for system privileges	User, Student, Parent, Admin, Tutor	Request for user information	User, Student, Parent, Admin, Tutor																																				
Name	Description																																																												
Name	Collaborator																																																												
Provide username	Login interface, Student, Parent, Admin,																																																												
Provide password	Login interface																																																												
Allow user to change password	Login interface																																																												
Logout	Login interface																																																												
Name	Description																																																												
Name	Collaborator																																																												
Checks username and password	User																																																												
Request for system privileges	User, Student, Parent, Admin, Tutor																																																												
Request for user information	User, Student, Parent, Admin, Tutor																																																												
<div>Student</div> <div>Super Classes: User</div> <div>Sub Classes:</div> <div>Description: Stores information about student user</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Retrieve username</td><td>User</td></tr><tr><td>Provide studentid</td><td></td></tr><tr><td>Provide number of booked sessions</td><td>Session</td></tr><tr><td>Provide payment details</td><td>Payment</td></tr><tr><td>Provide average rating</td><td>Session</td></tr><tr><td>Provide student system privileges</td><td></td></tr></table>	Name	Description	Name	Collaborator	Retrieve username	User	Provide studentid		Provide number of booked sessions	Session	Provide payment details	Payment	Provide average rating	Session	Provide student system privileges		<div>Parent</div> <div>Super Classes: User</div> <div>Sub Classes:</div> <div>Description: Stores information about parent user</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Retrieve username</td><td>User</td></tr><tr><td>Provide parentid</td><td></td></tr><tr><td>Provide list of child accounts</td><td>Student</td></tr><tr><td>Provide notification preferences</td><td></td></tr><tr><td>Provide payment details</td><td>Payment</td></tr><tr><td>Provide parent system privileges</td><td></td></tr></table>	Name	Description	Name	Collaborator	Retrieve username	User	Provide parentid		Provide list of child accounts	Student	Provide notification preferences		Provide payment details	Payment	Provide parent system privileges		<div>Admin</div> <div>Super Classes: User</div> <div>Sub Classes:</div> <div>Description: Stores information about admin user</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Retrieve username</td><td>User</td></tr><tr><td>Provide admin system privileges</td><td></td></tr></table>	Name	Description	Name	Collaborator	Retrieve username	User	Provide admin system privileges		<div>Tutor</div> <div>Super Classes: User</div> <div>Sub Classes:</div> <div>Description: Stores information about tutor user</div> <div>Attributes:</div> <table><tr><th>Name</th><th>Description</th></tr></table> <div>Responsibilities:</div> <table><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Retrieve username</td><td>User</td></tr><tr><td>Provide lists of subjects</td><td>Subject</td></tr><tr><td>Provide experience details</td><td></td></tr><tr><td>Provide schedule availability</td><td></td></tr><tr><td>Provide hourly rate</td><td></td></tr><tr><td>Provide average rating</td><td>Session</td></tr><tr><td>Provide tutor system privileges</td><td></td></tr></table>	Name	Description	Name	Collaborator	Retrieve username	User	Provide lists of subjects	Subject	Provide experience details		Provide schedule availability		Provide hourly rate		Provide average rating	Session	Provide tutor system privileges	
Name	Description																																																												
Name	Collaborator																																																												
Retrieve username	User																																																												
Provide studentid																																																													
Provide number of booked sessions	Session																																																												
Provide payment details	Payment																																																												
Provide average rating	Session																																																												
Provide student system privileges																																																													
Name	Description																																																												
Name	Collaborator																																																												
Retrieve username	User																																																												
Provide parentid																																																													
Provide list of child accounts	Student																																																												
Provide notification preferences																																																													
Provide payment details	Payment																																																												
Provide parent system privileges																																																													
Name	Description																																																												
Name	Collaborator																																																												
Retrieve username	User																																																												
Provide admin system privileges																																																													
Name	Description																																																												
Name	Collaborator																																																												
Retrieve username	User																																																												
Provide lists of subjects	Subject																																																												
Provide experience details																																																													
Provide schedule availability																																																													
Provide hourly rate																																																													
Provide average rating	Session																																																												
Provide tutor system privileges																																																													

Powered ByVisual Paradigm Community Edition

# TutorFlow

## 3.2.2 Search Subsystem



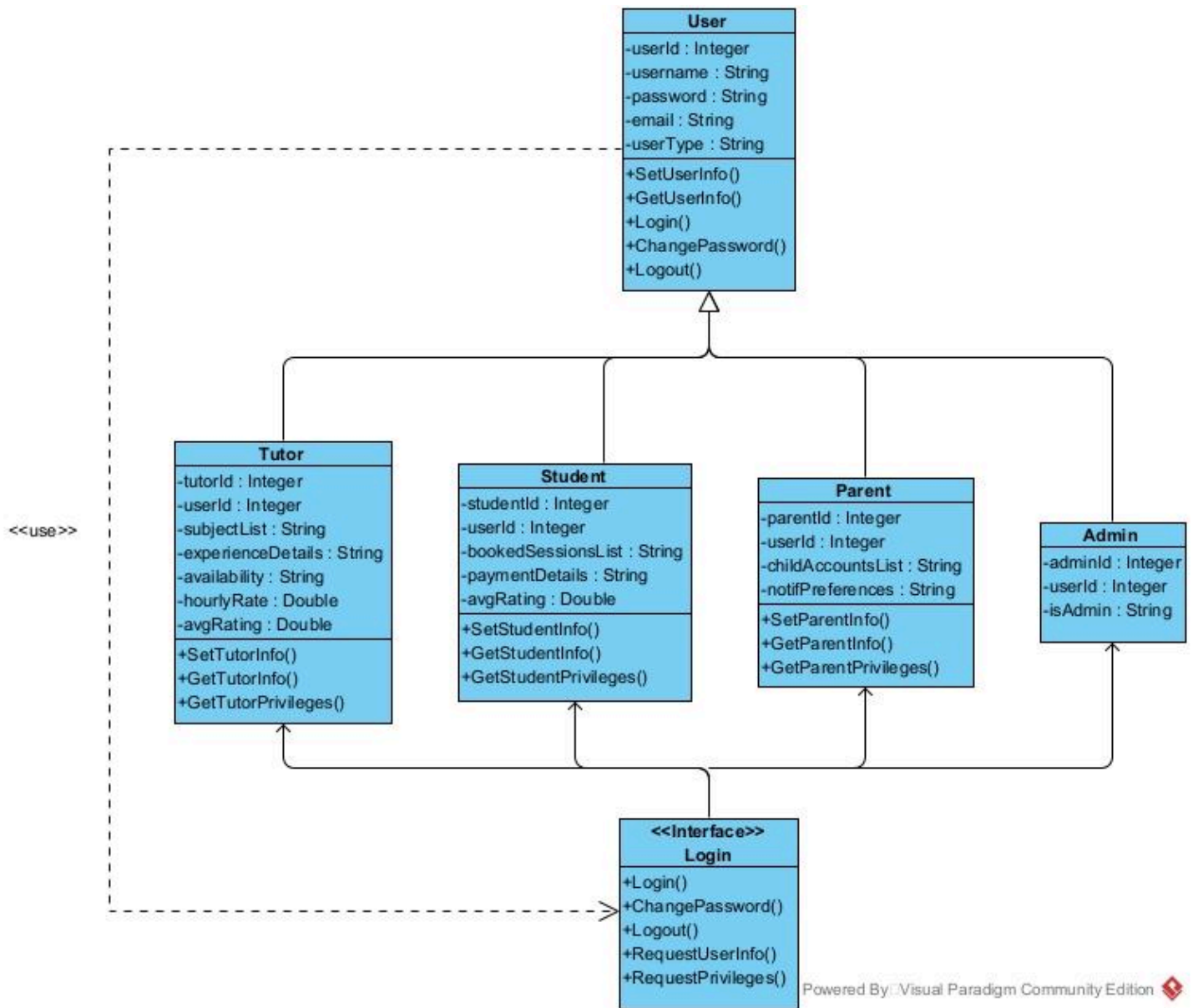
## TutorFlow

### 3.2.3 Rating/Feedback Subsystem

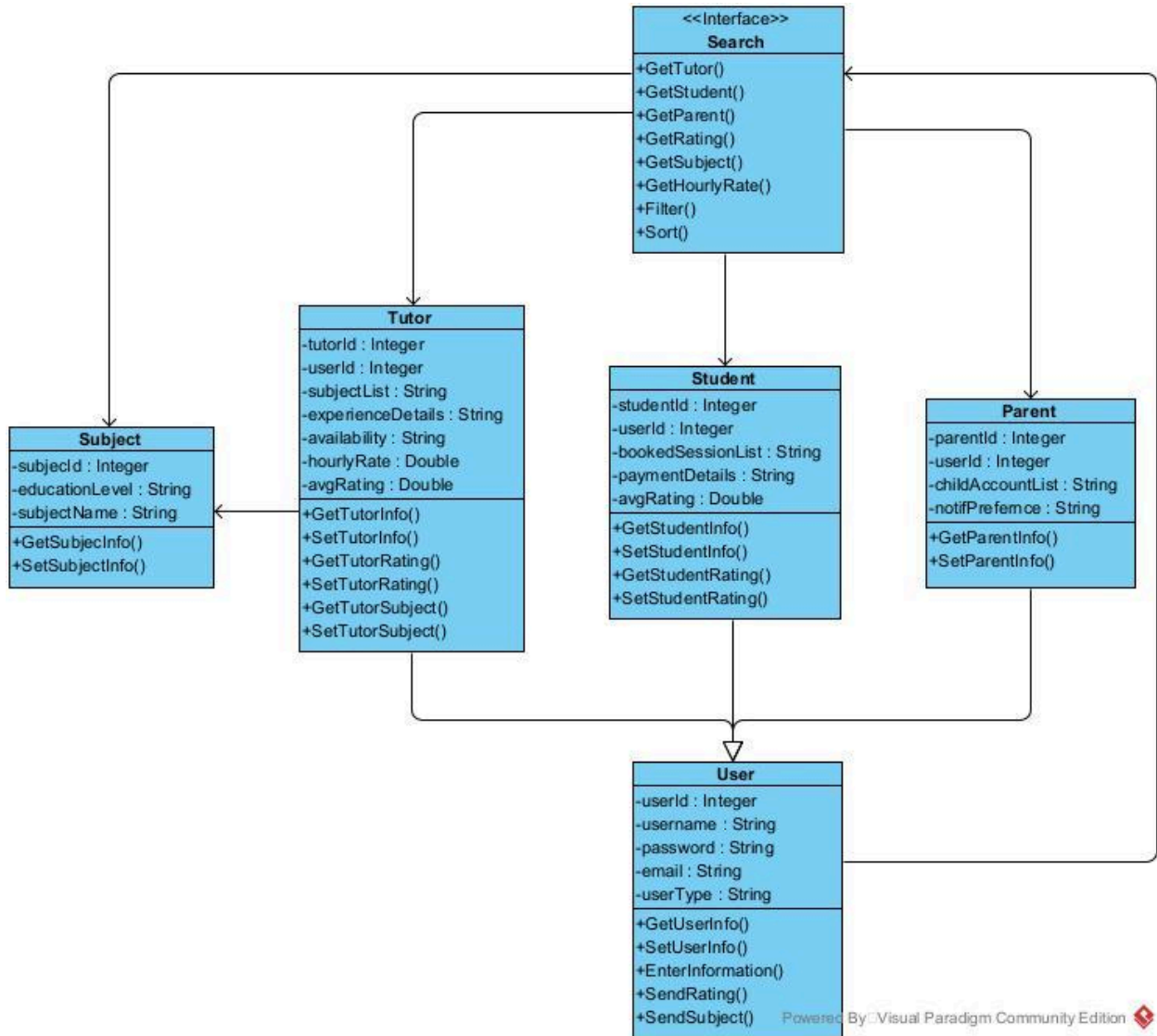
<div>User</div> <div><div>Sub Classes: Student, Parent, Tutor, Admin</div><div><div>Description: Manages the feedback and rating mechanism in the system, allowing users to provide feedback, rate sessions, and interact with these entries.</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Submit written feedback</td><td>Student, Parent, Tutor</td></tr><tr><td>Submit Ratings</td><td>Student, Parent, Tutor</td></tr><tr><td>Reply to Feedback</td><td>Tutor, Admin</td></tr><tr><td>Report Feedback</td><td>All users</td></tr><tr><td>Edit Feedback</td><td>All users</td></tr><tr><td>Delete Feedback</td><td>All users</td></tr><tr><td>Highlight Feedback</td><td>Tutor</td></tr><tr><td>Allow admin to action on inappropriate feedback</td><td>Admin</td></tr></table></div></div></div>	Name	Description	Responsibilities:		Name	Collaborator	Submit written feedback	Student, Parent, Tutor	Submit Ratings	Student, Parent, Tutor	Reply to Feedback	Tutor, Admin	Report Feedback	All users	Edit Feedback	All users	Delete Feedback	All users	Highlight Feedback	Tutor	Allow admin to action on inappropriate feedback	Admin	<div>Parent</div> <div><div>Sub Classes:</div><div><div>Description: Represents parents who provide feedback and rate sessions on behalf of their children.</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Check feedback given by or given to their child</td><td>Student, Feedback</td></tr><tr><td>Provide feedback on behalf of child</td><td>Student, Feedback</td></tr></table></div></div></div> <div>Tutor</div> <div><div>Sub Classes:</div><div><div>Description: Represents a tutor in the system who can give and receive feedback from students.</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Provide feedback for a session with a student</td><td>Feedback, Student, Session</td></tr><tr><td>Provide a rating for a student</td><td>Student</td></tr><tr><td>View feedback and ratings previously given by the tutor</td><td>Feedback</td></tr><tr><td>View feedback and rating provided by a student</td><td>Feedback, Student</td></tr></table></div></div></div>	Name	Description	Responsibilities:		Name	Collaborator	Check feedback given by or given to their child	Student, Feedback	Provide feedback on behalf of child	Student, Feedback	Name	Description	Responsibilities:		Name	Collaborator	Provide feedback for a session with a student	Feedback, Student, Session	Provide a rating for a student	Student	View feedback and ratings previously given by the tutor	Feedback	View feedback and rating provided by a student	Feedback, Student	<div>Admin</div> <div><div>Sub Classes:</div><div><div>Description: Represents an administrator who can oversee and manage the feedback system.</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Oversee all feedback given in the system for any violations or issues</td><td>Feedback, Student, Tutor</td></tr><tr><td>Take actions on feedback that violates platform rules</td><td>Feedback</td></tr></table></div></div></div> <div>Feedback</div> <div><div>Sub Classes:</div><div><div>Description: Manages individual feedback entries, recording details such as ratings and comments.</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Collect feedback from users</td><td>Student, Tutor, Parent</td></tr><tr><td>Provide a platform for users to view their received feedback</td><td></td></tr><tr><td>Allow reply to the feedback.</td><td>Student, Tutor, Parent</td></tr></table></div></div></div>	Name	Description	Responsibilities:		Name	Collaborator	Oversee all feedback given in the system for any violations or issues	Feedback, Student, Tutor	Take actions on feedback that violates platform rules	Feedback	Name	Description	Responsibilities:		Name	Collaborator	Collect feedback from users	Student, Tutor, Parent	Provide a platform for users to view their received feedback		Allow reply to the feedback.	Student, Tutor, Parent	<div>Notification</div> <div><div>Sub Classes:</div><div><div>Description: Manages the notification system, ensuring that users are informed about new ratings and feedback</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Send notifications about sessions</td><td>Student, Tutor, Parent</td></tr><tr><td>Alert users of new feedback/ratings</td><td>Student, Tutor, Parent</td></tr><tr><td>Inform tutors about booked sessions</td><td>Tutors</td></tr></table></div></div></div> <div>Session</div> <div><div>Sub Classes:</div><div><div>Description: Represents sessions between a student and a tutor</div><div>Attributes:<table><tr><th>Name</th><th>Description</th></tr><tr><td colspan="2">Responsibilities:</td></tr><tr><th>Name</th><th>Collaborator</th></tr><tr><td>Link feedback post-session</td><td></td></tr><tr><td>Take feedback and ratings for a given session</td><td>Student, Tutor, Parent</td></tr></table></div></div></div>	Name	Description	Responsibilities:		Name	Collaborator	Send notifications about sessions	Student, Tutor, Parent	Alert users of new feedback/ratings	Student, Tutor, Parent	Inform tutors about booked sessions	Tutors	Name	Description	Responsibilities:		Name	Collaborator	Link feedback post-session		Take feedback and ratings for a given session	Student, Tutor, Parent
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Submit written feedback	Student, Parent, Tutor																																																																																												
Submit Ratings	Student, Parent, Tutor																																																																																												
Reply to Feedback	Tutor, Admin																																																																																												
Report Feedback	All users																																																																																												
Edit Feedback	All users																																																																																												
Delete Feedback	All users																																																																																												
Highlight Feedback	Tutor																																																																																												
Allow admin to action on inappropriate feedback	Admin																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Check feedback given by or given to their child	Student, Feedback																																																																																												
Provide feedback on behalf of child	Student, Feedback																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Provide feedback for a session with a student	Feedback, Student, Session																																																																																												
Provide a rating for a student	Student																																																																																												
View feedback and ratings previously given by the tutor	Feedback																																																																																												
View feedback and rating provided by a student	Feedback, Student																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Oversee all feedback given in the system for any violations or issues	Feedback, Student, Tutor																																																																																												
Take actions on feedback that violates platform rules	Feedback																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Collect feedback from users	Student, Tutor, Parent																																																																																												
Provide a platform for users to view their received feedback																																																																																													
Allow reply to the feedback.	Student, Tutor, Parent																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Send notifications about sessions	Student, Tutor, Parent																																																																																												
Alert users of new feedback/ratings	Student, Tutor, Parent																																																																																												
Inform tutors about booked sessions	Tutors																																																																																												
Name	Description																																																																																												
Responsibilities:																																																																																													
Name	Collaborator																																																																																												
Link feedback post-session																																																																																													
Take feedback and ratings for a given session	Student, Tutor, Parent																																																																																												

### 3.3 TutorFlow Design Classes Diagram

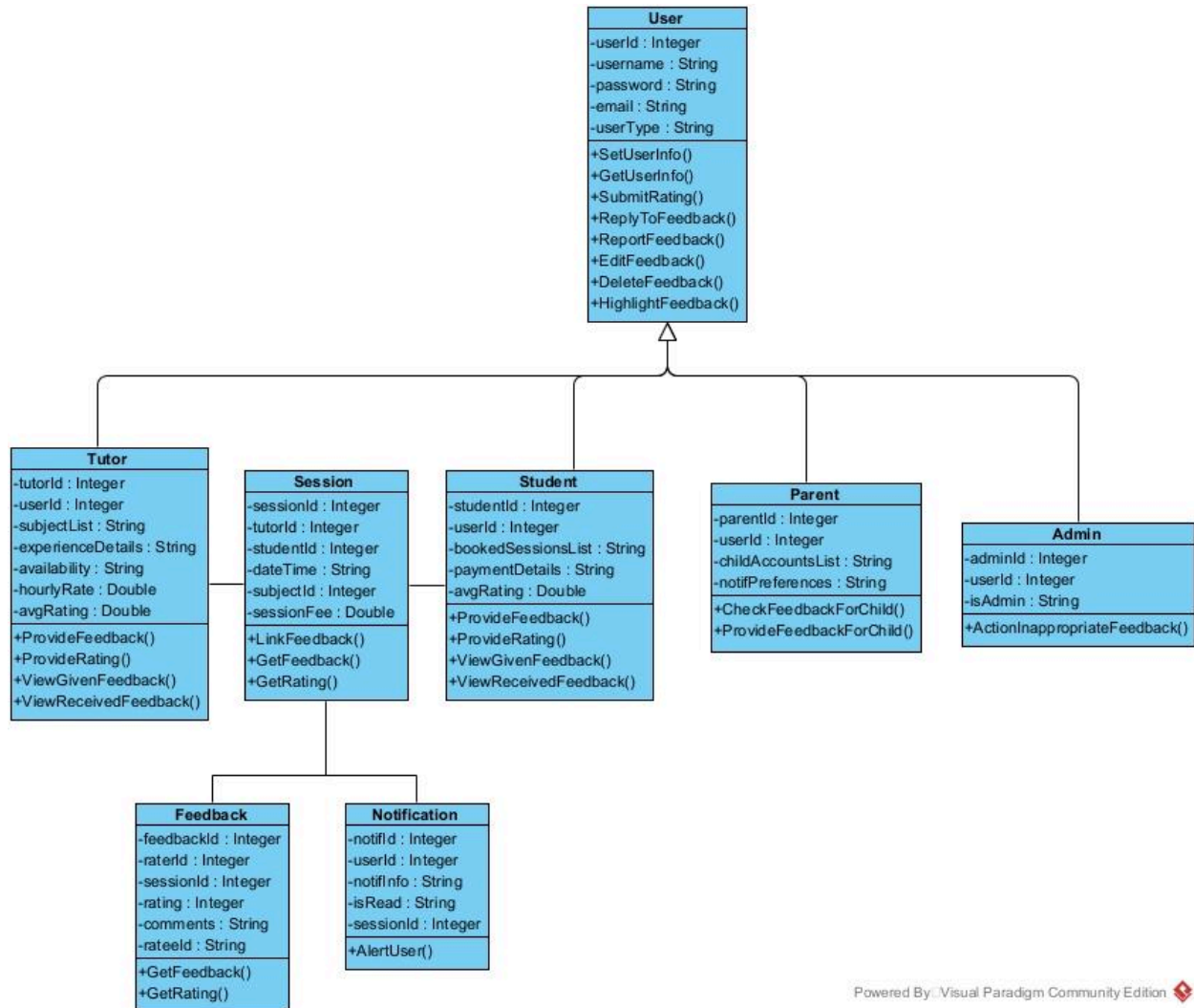
#### 3.3.1 Login Subsystem



### 3.3.2 Search Subsystem



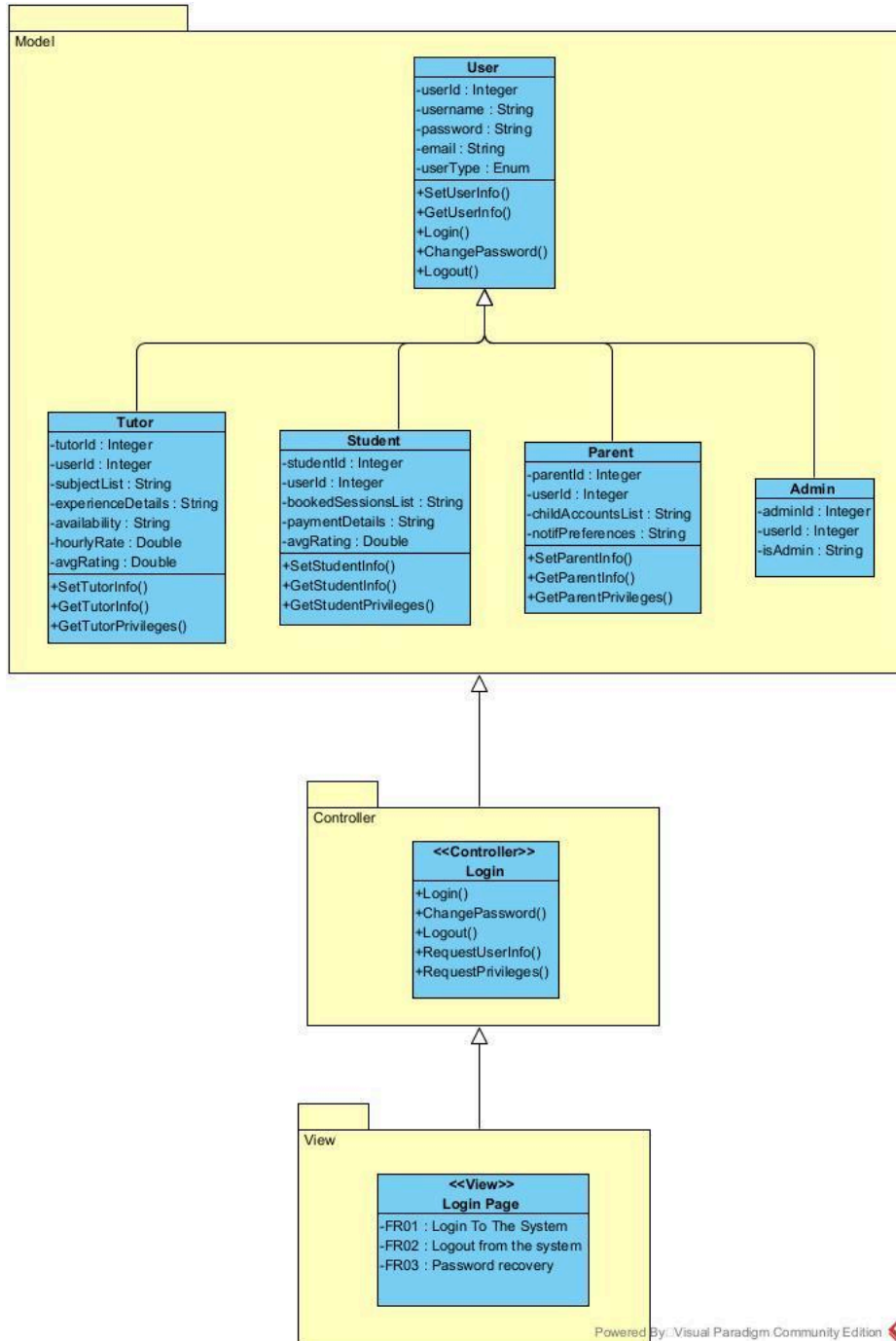
### 3.3.3 Rating/Feedback Subsystem



## Section 4: Framework M(odel) V(iew) C(ontroller)

### 4.1 MVC pattern diagram to include

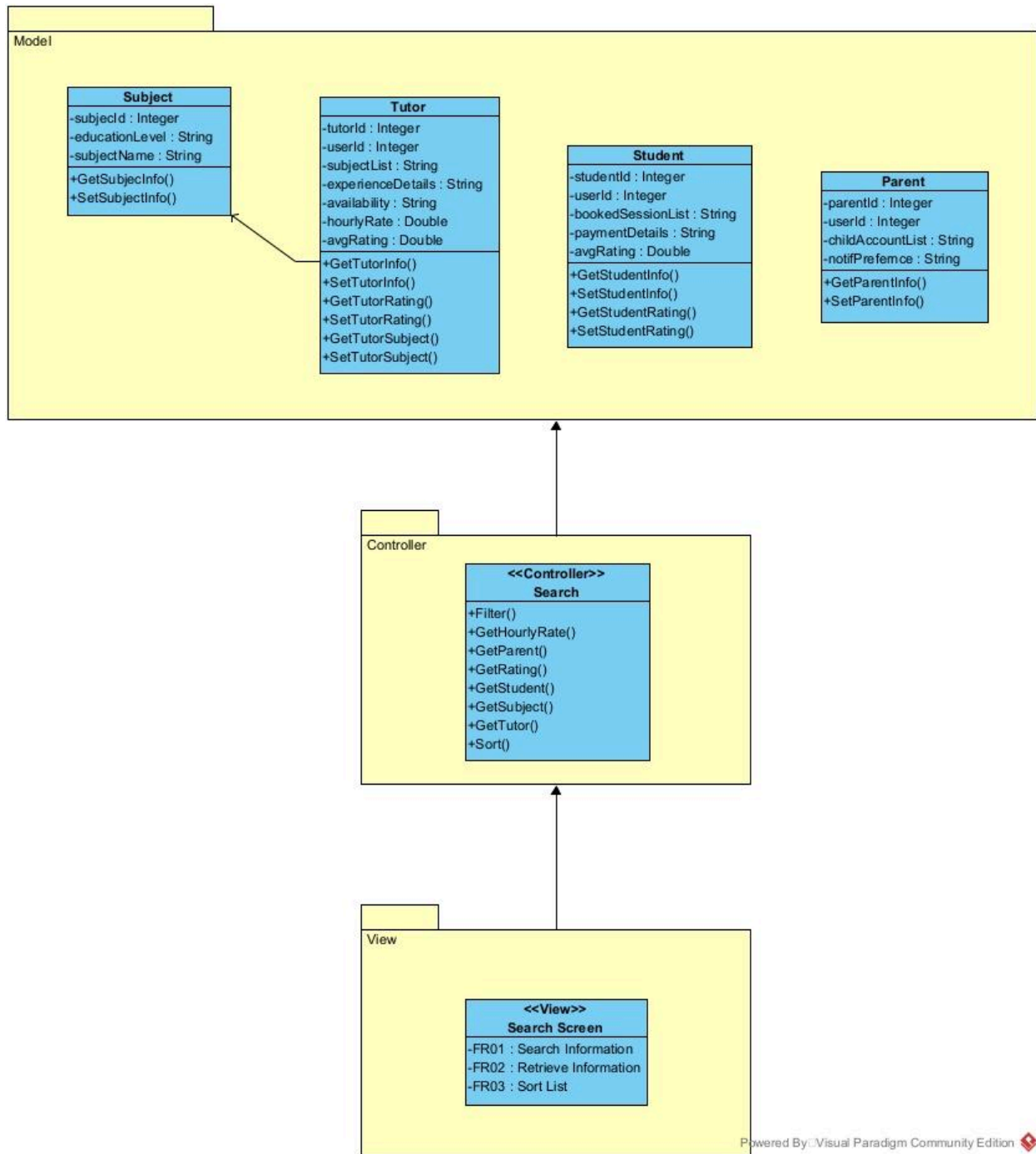
#### 4.1.1 Login Subsystem





# TutorFlow

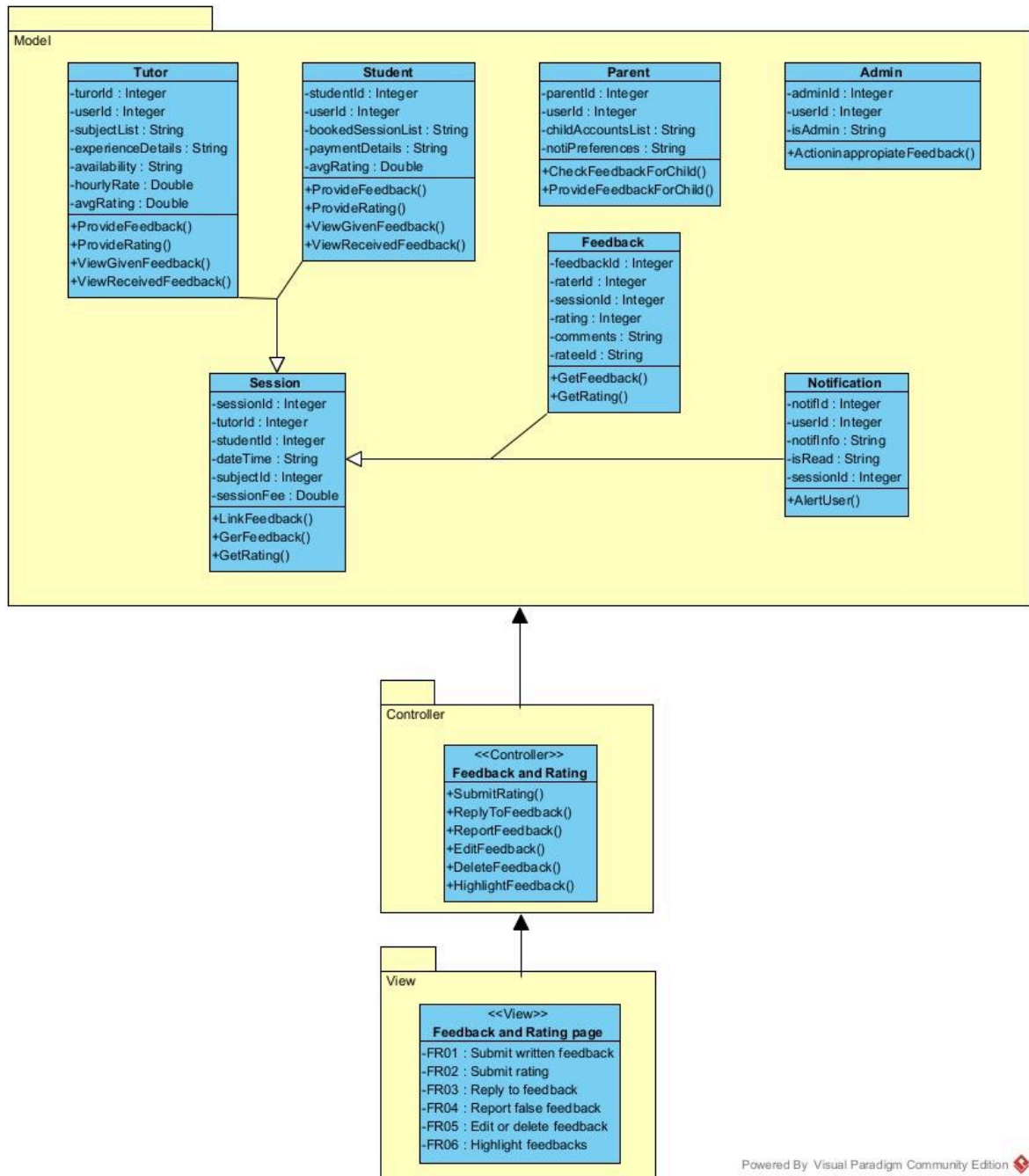
## 4.1.2 Search Subsystem





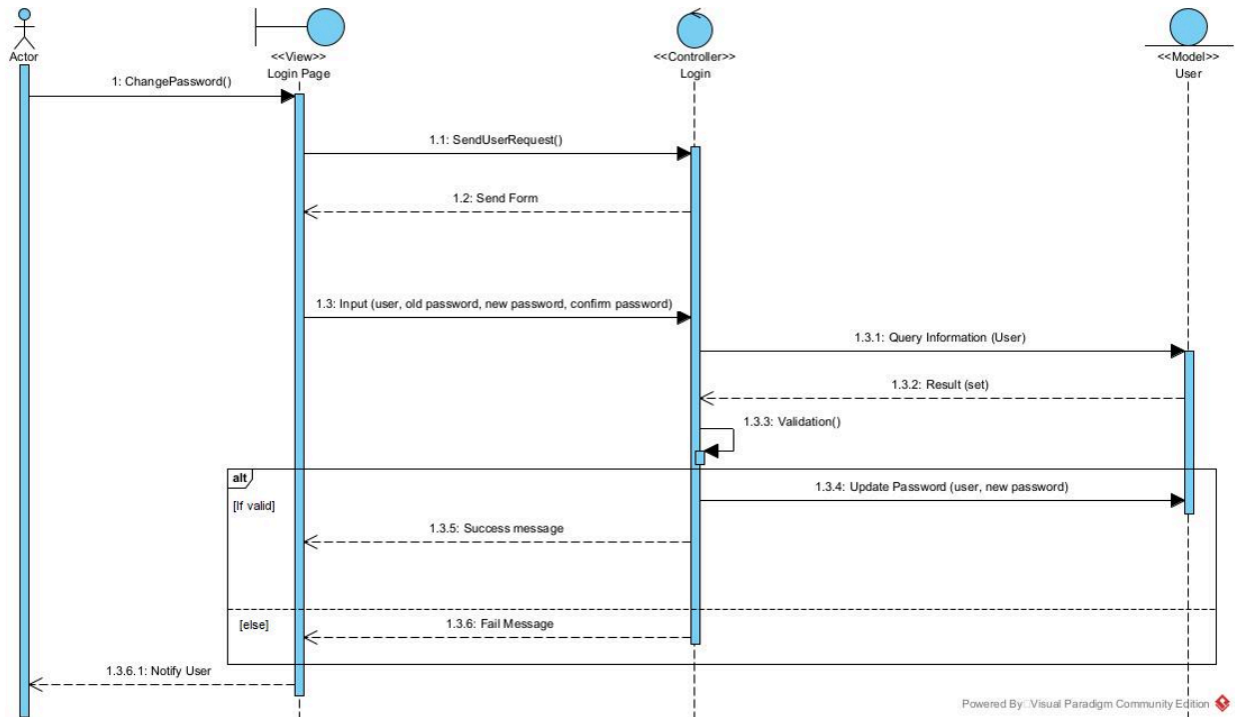
# TutorFlow

## 4.1.3 Rating/Feedback Subsystem

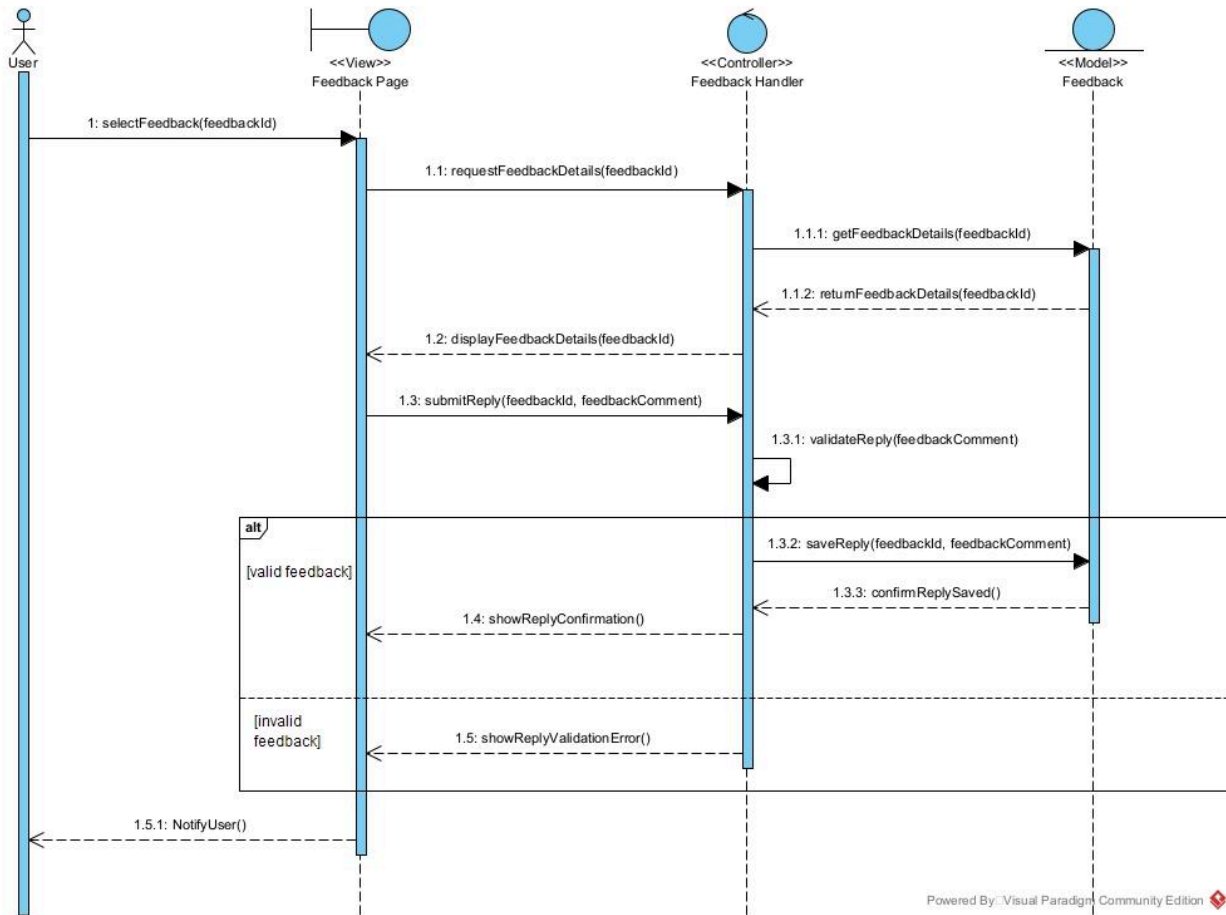


## 4.2 Full Sequence Diagrams

### 4.2.1 Use case: Update Password (Login Subsystem)

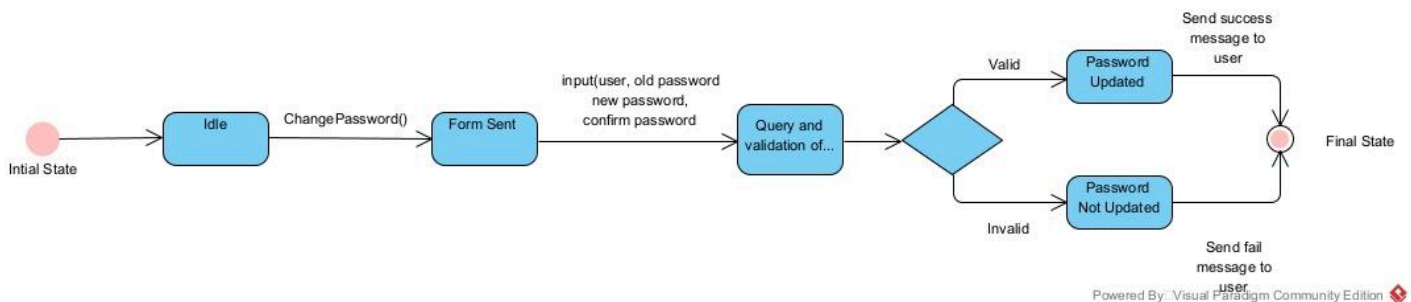


## 4.2.2 Use case: Replying to feedback (Rating/Feedback Subsystem)

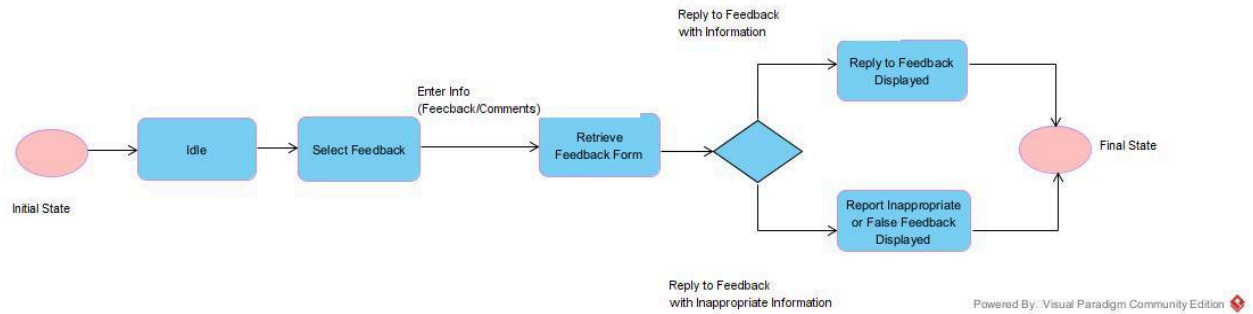


## 4.3 State Machine Diagrams

### 4.3.1 Use Case: Update Password [for User and Login Interface objects]

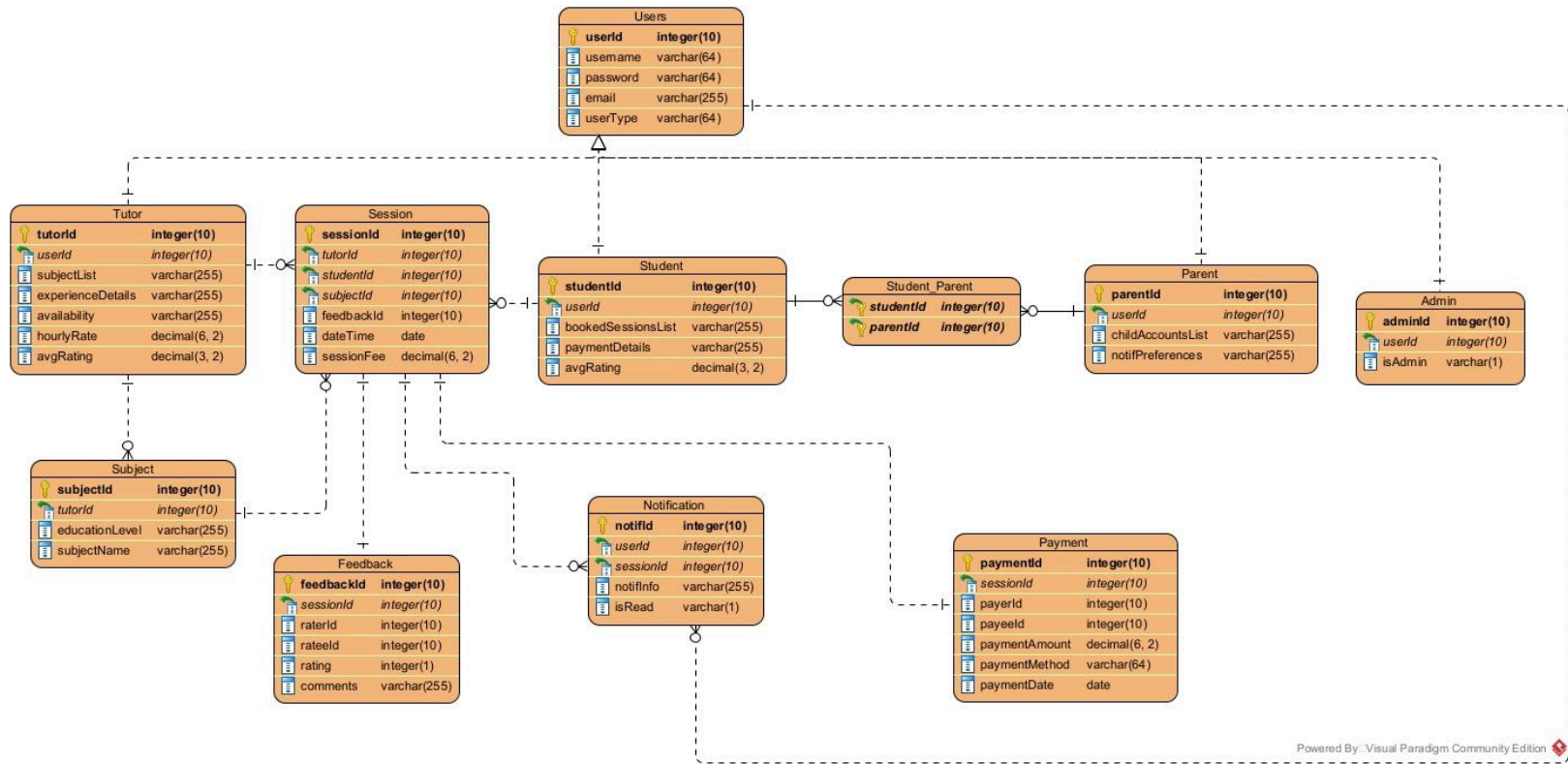


## 4.3.2 Use Case: Replying to feedback [for Tutor and Feedback objects]



## Section 5: Data Layer

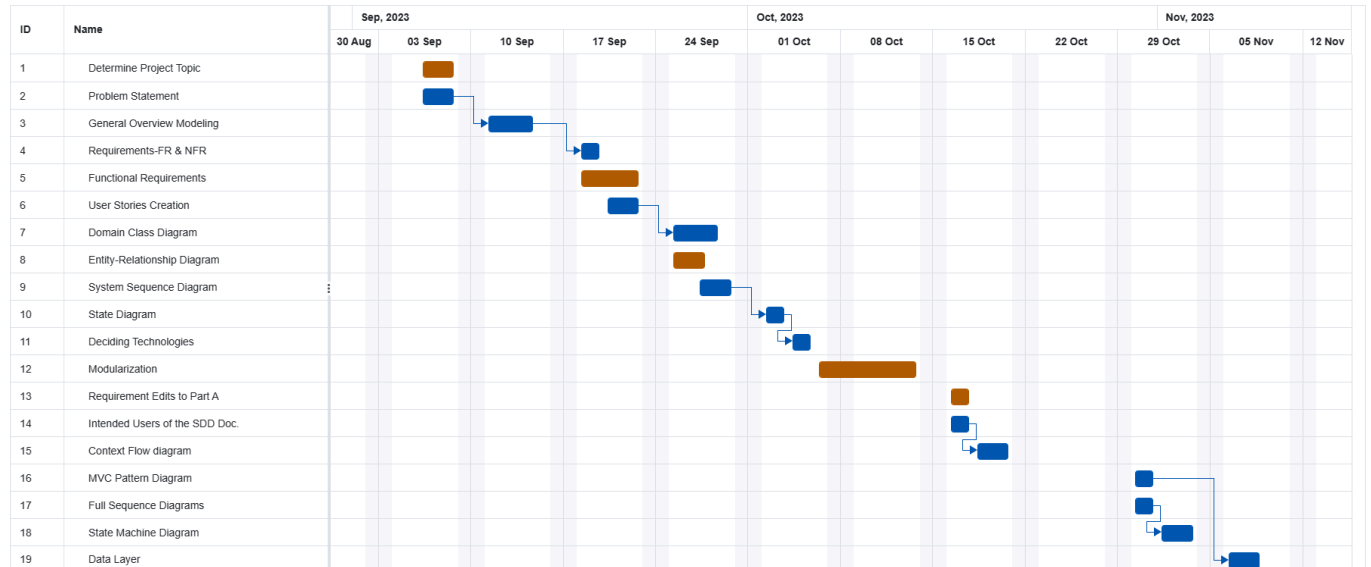
### 5.1 Database Schema



## 5.2 Technology List Update

No technology updates are required.

## Section 6: Gantt chart update



## Part C: System Design Documents

### Section 1: Corrections to Design Specifications Part B

No corrections had to be made to Part B.

### Section 2: Software Design Patterns

#### 2.1 Observer Pattern

**Explanation:**

- The observer pattern can be used to establish a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- This pattern is useful when changes to one object require changes to another, especially in a scenario where manual updates are not feasible.

**Application:**

- In the context of TutorFlow, the observer pattern can be used to notify users when new feedback is available or if a reply to their feedback has been posted. The feedback class will be the subject, while the notification class and the user interface that displays feedback will be the observers.

#### 2.2 Facade Pattern

**Explanation:**

- The Facade pattern provides a simplified interface to a set of interfaces in a subsystem.
- It defines a higher-level interface that makes the subsystem easier to use. The Facade pattern is often used to simplify complex systems by providing a single point of interaction rather than exposing the underlying complexity to the client.

**Application:**

- In the context of TutorFlow, the facade pattern can be used to create a simpler interface for the complex MVC layers in the system. For example, a FeedbackManagementFacade could provide simple methods to post feedback, reply to feedback, and retrieve feedback without exposing the underlying complexities of the database and business logic layers.

## 2.3 Adapter Pattern

### Explanation:

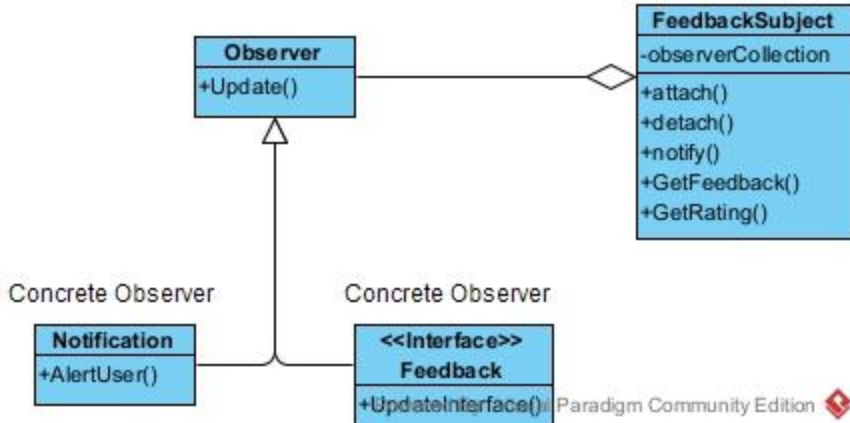
- The Adapter pattern allows for interfaces of different classes to work together. It converts the interface of a class into another interface clients expect. The Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

### Application:

- In the context of TutorFlow, the adapter pattern could be beneficial when integrating with OAuth providers with social logins such as Google, Facebook or Twitter. These services have their own authentication interfaces and protocols, which may not directly align with TutorFlow's login interface. Another example is the integration of payment systems necessary for users to send money for sessions booked.

## Section 3: Using common software design patterns

<b>Software Design Pattern:</b>	Observer
<b>Problem:</b>	In TutorFlow, there is a need to automatically inform users about updates regarding feedback on their sessions. Manually notifying each user about new feedback or replies could be inefficient and prone to errors.
<b>Solution:</b>	The solution to this problem is to implement the Observer pattern by designating the Feedback class as the subject that maintains a list of observers, which includes the Notification class and the user interface components that display feedback information. Whenever a change occurs in the Feedback (such as receiving new feedback or a reply), it will notify all registered observers related to the new feedback, triggering them to update their view or state accordingly.

<b>Graph:</b>	 <pre> classDiagram     class Observer {         +Update()     }     class FeedbackSubject {         -observerCollection         +attach()         +detach()         +notify()         +GetFeedback()         +GetRating()     }     class Notification {         +AlertUser()     }     class Feedback {         &lt;&lt;interface&gt;&gt;         +UpdateInterface()     }     Observer &lt; -- ConcreteObserver     Observer o--&gt; FeedbackSubject     Notification --&gt; ConcreteObserver     FeedbackSubject o--&gt; Feedback     </pre> <p>The diagram illustrates the Observer pattern. It features an <b>Observer</b> interface with a <code>+Update()</code> method. A <b>Concrete Observer</b> class implements this interface. A <b>Notification</b> class has a <code>+AlertUser()</code> method that triggers the update process. The <b>FeedbackSubject</b> class maintains a collection of <b>Observer</b> objects (indicated by a diamond on the association line) and provides methods to manage them: <code>+attach()</code>, <code>+detach()</code>, <code>+notify()</code>, <code>+GetFeedback()</code>, and <code>+GetRating()</code>. A <b>Feedback</b> interface defines the <code>+UpdateInterface()</code> method, which the <b>FeedbackSubject</b> implements. The diagram is labeled 'Concrete Observer' for both the <b>Notification</b> and <b>Feedback</b> classes. A 'Paradigm Community Edition' logo is visible in the bottom right corner of the diagram area.</p>
<b>Benefits:</b>	<p>Automatic updates enhance the user experience by ensuring timely notifications without manual intervention. It also reduces the risk of missed notifications and ensures all observers are updated simultaneously, thus reducing the errors of this subsystem.</p>
<b>Consequences:</b>	<p>The implementation of the Observer pattern may introduce complexity if not managed well. This is especially likely if there are many observers and notifications need to be sent in real-time. Observers must be managed to prevent memory leaks, and care must be taken to ensure that notifications do not become a performance bottleneck for the rest of the subsystems and the platform as a whole.</p>



## Section 4: UX/UI Design

### 4.1 Login Page

TutorFlow

HomePricingAboutContact

TutorFlow

Unlock Your Potential with TutorFlow: Where Learning Flows Seamlessly.

Log in to TutorFlow

Email or username

Password

Forgot your password?

LOGIN

New here? [Create an account](#)

9:41

tutorflow.ca

# TutorFlow

Unlock Your Potential with TutorFlow:  
Where Learning Flows Seamlessly.

## Log in to TutorFlow

[Forgot your password?](#)

LOG IN

[New here? Create an account](#)

## 4.2 View Feedback and Ratings for a Tutor

TutorFlow

[Home](#)[Pricing](#)[About](#)[Contact](#)

8/10 Overall Quality Based on 19 Ratings

★★★★★☆☆

PSVCH101PSVCH220PSVCH231

Kara Danvers

Professor in the Psychology department  
Lorem ipsum dolor sit amet consectetur. Nunc urna nullam sed pulvinar semper porta sapien nec. Sit tortor malesuada egestas parturient.

Rate →

All Courses

Most Recent

★ Highlighted review

Jeff

parent account

Apr 7th, 2011

PSVCH101

★ 10/10

Mollis vel lorem curabitur ultrices. Ullamcorper nunc arcu donec feugiat. Lectus ut feugiat eu integer luctus aliquam ridiculus placerat vivamus. Enim dui suscipit morbi hac libero phasellus tortor. Mauris vulputate ut volutpat quis habitant hendrerit eleifend elementum. Congue adipiscing massa ut in. Molestie ullamcorper tincidunt eget arcu velit neque aenean suspendisse. Et amet nunc facilisis ornare. Diam platea venenatis eget dolor tincidunt. Ultrices dignissim in massa sit rutrum. Eu nunc lorem leo tortor ipsum sed vulputate. Etiam lacus diam non erat dignissim amet odio dolor dictum. Viverra volutpat vulputate etiam non.

Auli

student account

Apr 7th, 2011

PSVCH101

★ 7/10

Mollis vel lorem curabitur ultrices. Ullamcorper nunc arcu donec feugiat. Lectus ut feugiat eu integer luctus aliquam ridiculus placerat vivamus. Enim dui suscipit morbi hac libero phasellus tortor. Mauris vulputate ut volutpat quis habitant hendrerit eleifend elementum. Congue adipiscing massa ut in. Molestie ullamcorper

9:41

tutorflow.ca

TutorFlow

8 /10

Overall Quality Based on 19 Ratings

PSYCH101

PSYCH220

PSYCH231

Kara Danvers

Professor in the Psychology department  
Lorem ipsum dolor sit amet consectetur. Nunc urna nullam sed pulvinar semper porta sapien nec. Sit tortor malesuada egestas parturient.

Rate →

All Courses

Most Recent

★ Highlighted review

PSYCH101

★ 10/10

Jeff

parent account

Apr 7th, 2011

Mollis vel lorem curabitur ultrices. Ullamcorper nunc arcu donec feugiat. Lectus ut feugiat eu integer luctus aliquam ridiculus placerat vivamus. Enim dui suscipit morbi hac libero phasellus tortor. Mauris vulputate ut volutpat quis habitant hendrerit eleifend elementum. Congue adipiscing massa ut in. Molestie ullamcorper tincidunt eget arcu velit neque aenean suspendisse. Et amet nunc facilisis ornare. Diam platea venenatis eget dolor tincidunt. Ultrices dignissim in massa sit rutrum. Eu nunc lorem leo tortor ipsum sed vulputate. Etiam lacus diam non erat dignissim amet odio dolor dictum. Viverra volutpat vulputate etiam non.

PSYCH101

★ 7/10

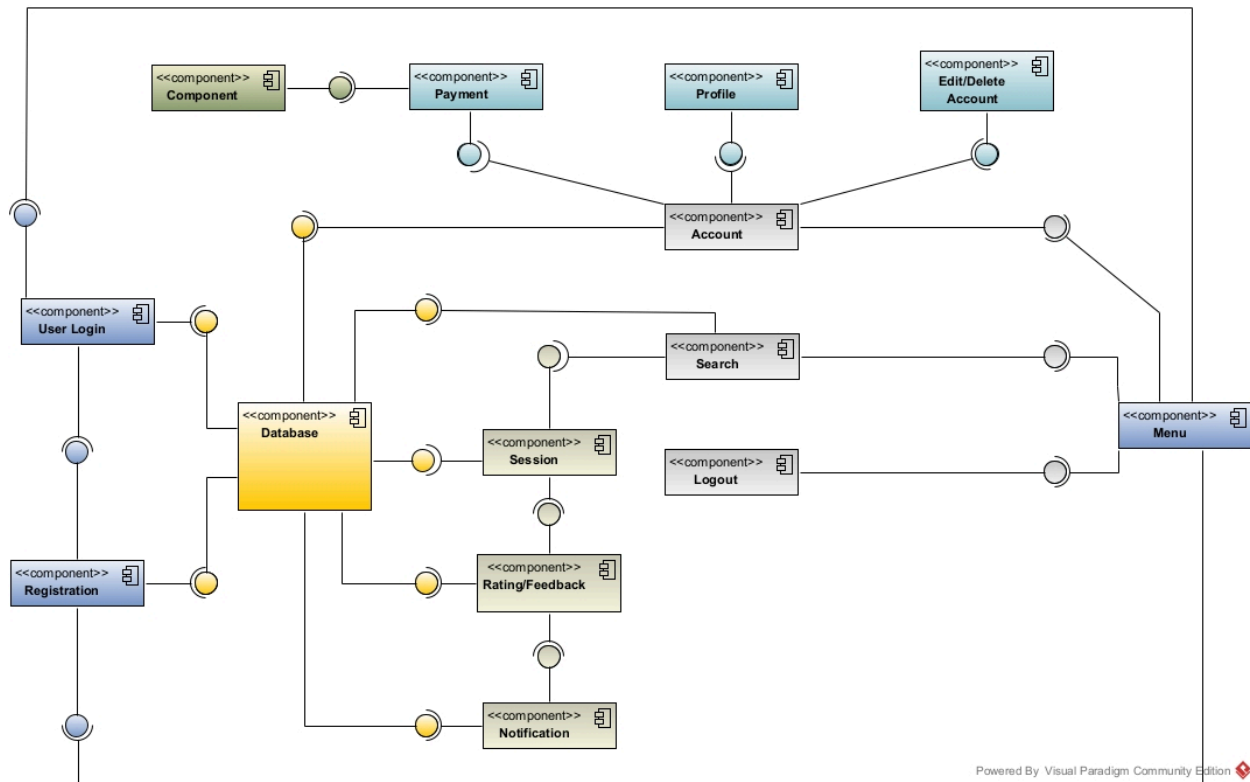
Auli

student account

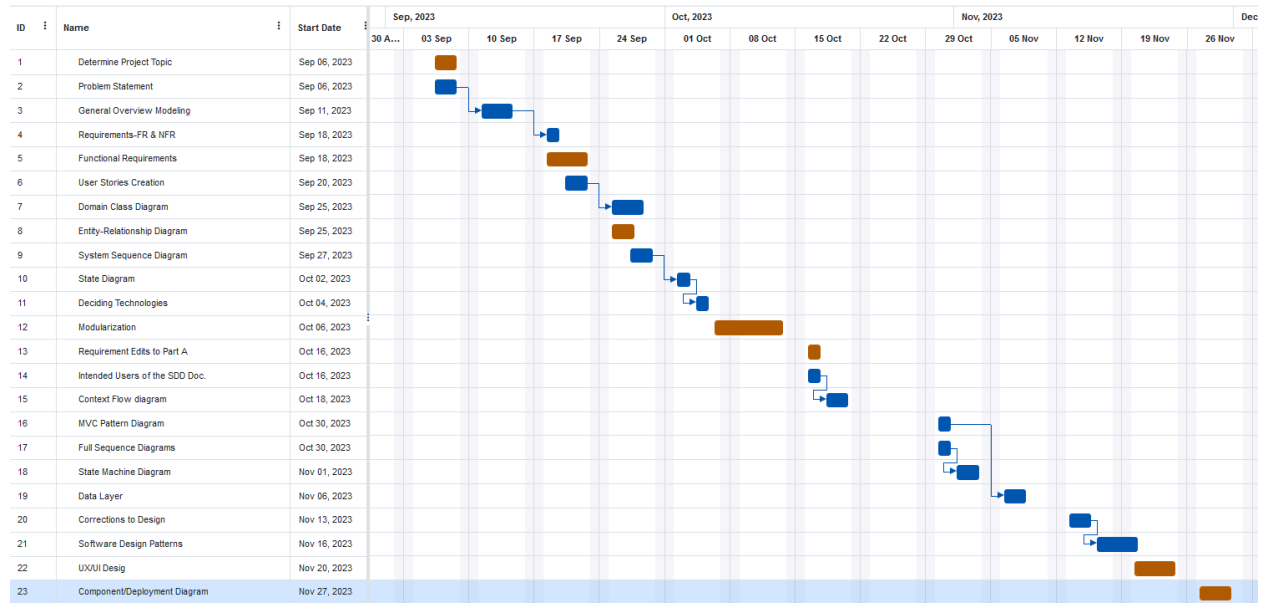
Apr 7th, 2011

Mollis vel lorem curabitur ultrices. Ullamcorper nunc arcu donec feugiat. Lectus ut feugiat eu integer luctus aliquam ridiculus placerat vivamus. Enim dui suscipit morbi hac libero phasellus tortor. Mauris vulputate ut volutpat quis habitant hendrerit eleifend elementum. Congue adipiscing massa ut in. Molestie ullamcorper tincidunt eget arcu velit neque aenean suspendisse. Et amet nunc facilisis ornare. Diam platea venenatis eget dolor tincidunt. Ultrices dignissim in massa sit rutrum. Eu nunc lorem leo tortor ipsum sed vulputate. Etiam lacus diam non erat dignissim amet odio dolor dictum. Viverra volutpat vulputate etiam non.

## Section 5: High level Component/Deployment Diagram



## Section 6: Update the Gantt chart to include Part C Tasks



## Section 7: Project Presentation

COMP 246 - TutorFlow