SE: 505 Software Project Lab II

"TUTORS PLANET"

SOFTWARE REQUIREMENT SPECIFICATION & ANALYSIS

Submitted by:

Rabiul Islam Sujon (Exam Roll: 1004)

Md. Nadim Ahmed (Exam Roll: 1009)

Supervised by:

Kishan Kumar Ganguly

Lecturer at IIT, University of Dhaka

SUBMISSION DATE: OCTOBER 22, 2020

Institute Of Information Technology, University of Dhaka

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter is a part of our software requirement specification for the project "Tutors Planet". In this chapter, we focus on the intended audience for this project.

1.1 Purpose

This document briefly describes the Software Requirement Analysis of "Tutors Planet". It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer. The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system. The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.

The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements. The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run

their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

CHAPTER 2

INCEPTION OF "TUTORS PLANET"

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

To establish the groundwork, the following factors have been worked on to the inception phase:

- Icebreaking
- List of stakeholders.
- Recognizing multiple viewpoints.
- Working towards collaboration.
- Requirements questionnaire.

2.2.1 Icebreaking

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this by talking with them in context-free languages. Their behavior, responding to our questions or willing to take a change in their shops solely depends on this phase.

2.1.2 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that might be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with some tutors and guardians and found some requirements.

We identified the following stakeholders for our “Tutors Planet” project.

- Tutor
- Guardian

2.1.3 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Tutor’s viewpoint:

- Easy to find tuition.
- verify tutors educational identity strongly.
- Manage Batch or Group

Guardian’s viewpoint:

- Easy to find tutor
- Easy to use
- Give post for tutor

2.1.4 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholders. That's why we followed the following steps to merge these requirements-

- ❖ Find the common and conflicting requirements.
- ❖ Categorize them.
- ❖ List the requirements based on stakeholder's priority points.
- ❖ Make final decisions about requirements.

Common viewpoints:

- Error free effective system
- User friendly
- Easy to maintain the software
- Strong authentication system.

Conflicting viewpoints:

- Creating a group management system.
- Developing the project in minimum budget

Final requirements:

We finalize the following requirements based on stakeholder's priority point:

- ❖ User friendly system.

- ❖ Strong authentication.
- ❖ Maximum error free system. (5%-10% error is considerable).
- ❖ Restrict access to functionality of the system based upon user roles.
- ❖ Easy to maintain the software.
- ❖ Adding some extra features like group management and messagebox, calendar, notes .

2.1.5 Requirements Questionnaire

At first some context free questions were asked for identifying the stakeholders. Context free questions are helpful to identifying some stakeholders who cannot be identified by structural questions. Then questions regarding the software were regarding their demands. The questionnaires are included in the appendix section.

2.2 Conclusion

The Inception phase helped us to establish a basic understanding about the tuition management system, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

CHAPTER 3

ELICITATION OF “TUTORS PLANET”

This chapter specifies the Elicitation phase.

3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

We have met with some stakeholders in the Inception phase such as the tutor, guardian. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are acting a vital role in the whole process) again to elicit the requirements.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximization. In order to ensure this, QFD enforces an understanding of what customers describe as ‘valuable’ and then deploy these values throughout the engineering process.

QFD defines three types of requirements:

Normal Requirements

Expected Requirements

Exciting Requirements

3.2.2.1 Normal Requirements

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for our project.

- Simple and user friendly interface
- Easily accessible for all
- Smartphone based application(Android)
- Allow new members to register.
- Login system.
- Providing the feature to post to find the tutor.
- Find tuition from all tuition posts.
- View tutor information and preferences
- Search for a tutor.

- A group and a batch management system.
- Providing a messagebox.
- Notifying members via SMS and email.
- Rate tutor.
- Block and report tutor and guardian.

3.2.2.2 EXPECTED REQUIREMENTS

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite not being explicitly mentioned, their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below:

- Database.
- Login type(tutor, guardian, admin)
- Error free software
- User friendly
- Data backup.
- Interactive and attractive graphical user interface.
- Strong authentication process.

3.2.2.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project.

- Users can create Google calendar events in the Calendar.
- Users can use the message box to communicate with guardians/tutors.

- Tutors can upload demo videos and guardians can understand the tutor's teaching method by watching demo videos.
- Tutors can create notes in the app.

3.2.3 Usage Scenario

1. Registration:

A) Guardian/Student's Account Creation: To use the system, the guardian/student who wants a tutor will open an account. Guardian/student will provide a username, area address (e.g.Azimpur), mobile number and an email (optional). A verification code will be sent in the guardian/student's phone number to verify the mobile number. If the user enters the code correctly, the account creation will be completed.

B) Tutor's Account Creation:

To open an account, firstly tutor will provide username, password, area address (e.g.Azimpur), gender, educational institution's name, subject's name (for university students), current position (e.g. 2nd year), email address (optional) and mobile number. A verification code will be sent in the tutor's email to verify the email.

Secondly, the authentication of a tutor will be checked whether he or she is a student of the educational institution that has been mentioned in the account. Here, the tutor will provide his institution's identity card's picture and will give 3 verified tutor's references. References will be referred to verified tutors by their email address. So, the tutor has to give the email address of the verified tutors from which he/she wants references. If the ID card is valid and the verified tutors approve the reference, anyone of the admin panel will approve the tutor account as a verified tutor. After this, the tutor account creation will be completed successfully. Other users can report about a tutor's fake information, and if anyone of the admins finds it as true, they can block the user.

2. Profile: There will be a profile of each guardian and verified tutor in the system. In the profile, a user can add a profile picture. Otherwise, a default picture will be used as a profile picture. The profile will show the username, profile picture, area address for each user. In the tutor's profile, the tutor's educational institution's name, email address (optional), mobile number (optional), gender, current position, subject (e.g. Math) will be added along with the previous information. In the profile, the tutor also can set the preferred medium of version (Bangla/English Medium or both), preferred classes (e.g. 11-12) and preferred subjects (e.g. Physics, Math), experience status if the tutor wants. Besides, a tutor can set different areas for

tuition in his/her preferred region (e.g. Jatrabari, Shahbag). If tuition is available for the tutor's preferred places, a notification will be sent to the tutor's profile.
Users can edit his/her profile information at any time except the educational institution's name and subject. But users can add new educational institution's names and subjects with authentication. There will be a Calendar, a Reminder and a Note as utility features in the tutor profile. Here, a tutor can mark a date, unmark a date, set a reminder in Calendar. Tutors can make notes in Note.

3. Demo: Tutors can upload one or more demo videos on his/her profile which can be viewed by others. Guardian/students can get some ideas about the teaching method of the tutor by watching those videos.

4. Post: Guardian can create a post for tuition to find a tutor with the following information of them:

- Post Title(optional)
- Student's institution name(optional)
- Class of the student (e.g. 11-12)
- Medium (Bangla/English)
- Tutor gender preferences (Male/Female)
- Subjects List
- Days per week/month
- Area address
- Contact number (optional)
- Tutor's educational institution name (optional)
- Tutor's subject name (optional)
- Salary range (optional)
- Others (optional)

The interested tutors can communicate with the guardian by using the message box. A notification will be sent to the guardian's profile when a tutor shows interest in the tuition. This notification can be controlled if a user wants. But, by default the notification system will be on. The guardian can remove the post at any moment. If the guardian finds out a desired tutor, he/she will set the post as unavailable. As a reminder, after 24 hours an app notification will go to the guardian's profile for setting up the tuition offer whether the tuition is available or not.

5. Searching and Viewing: Guardian can search the tutors based on the tutor's name, tutor's subject (e.g. Pharmacy), educational year, and educational institution and view the profile of the tutors. The Guardian also can search nearby Groups. The Guardian can communicate with the tutor via message box.

6. Message Box: There will be a message box for each user. If anyone wants to communicate with another user via message box, he/she will send an initial message request to another. If the other user accepts his/her request, then they can communicate with each other via message box. There will be an option for blocking a person. If a user seems to be bothering him/her, the user can block that person by which he or she will not be able to disturb the user.

7. Group Creation: A tutor can create a group where he/she can add one or more tutors. The group creator can edit group info (name, address, and image) and add or remove tutors. In a group, the group creator can create one or more batches. All information about the batch like academic days, schedule, payment, seat's availability will be mentioned. Batch info can be added, removed or edited by the group creator. There will be a StudentInfo to store information about the students. For instance, the name, address, phone number, institution's name, class of the students will be kept in the StudentInfo. There will also be a message box so that the guardians can communicate with the group. A notice board will be provided in the group so that any notice or post can be published about the group. The group creator can post on the notice board. The group creator can add attachment or files in the notice board. The guardians can view the posts and notices of the notice board.

3.2.4 Elicitation Work Product

At first, it has to be known whether the output of the Elicitation task may vary because of the dependency on the size of the system or the product to be built. Here, the Elicitation work product includes: -

- Making a bounded statement of scope for our system.
- Making a list of customers, users and other stakeholders who participated in the requirements elicitation.
- Making a list of requirements that are organized by function and domain constraints that apply to each other.
- A set of usage scenarios that provide insight into the use of the system.
- Description of the system's technical environment.

CHAPTER 4

SCENARIO BASED MODELING

This chapter describes the Scenario Based Model for the Cafeteria Management System.

4.1 Introduction

When developing software, user satisfaction is given the highest priority. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Thus, requirements being with scenario generation in the form of use cases, activity diagrams and swim lane diagrams.

4.2 Definition of use case

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

Primary Actor:

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor:

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.3 Use case diagrams

Use case diagrams give the non-technical view of the overall system.

1. Use case diagram level: 0

ID: L-0
Name: TutorsPlanet
Primary Actor: Tutor, Guardian, Admin

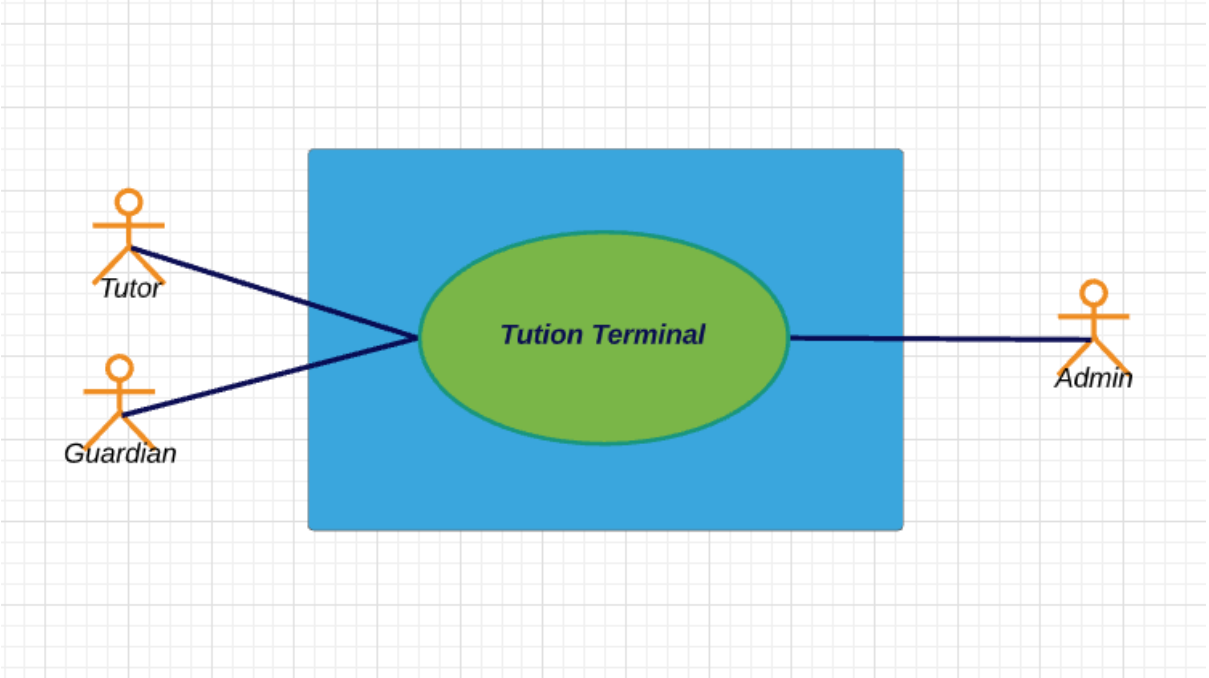


Figure 1: Use case diagram-1

Description: This use case shows the low level interaction between TutorsPlanet system and actors.

2. Use case diagram level: 1

ID: L-1
Name: TutorsPlanet
Primary Actor: Tutor, Guardian, Admin

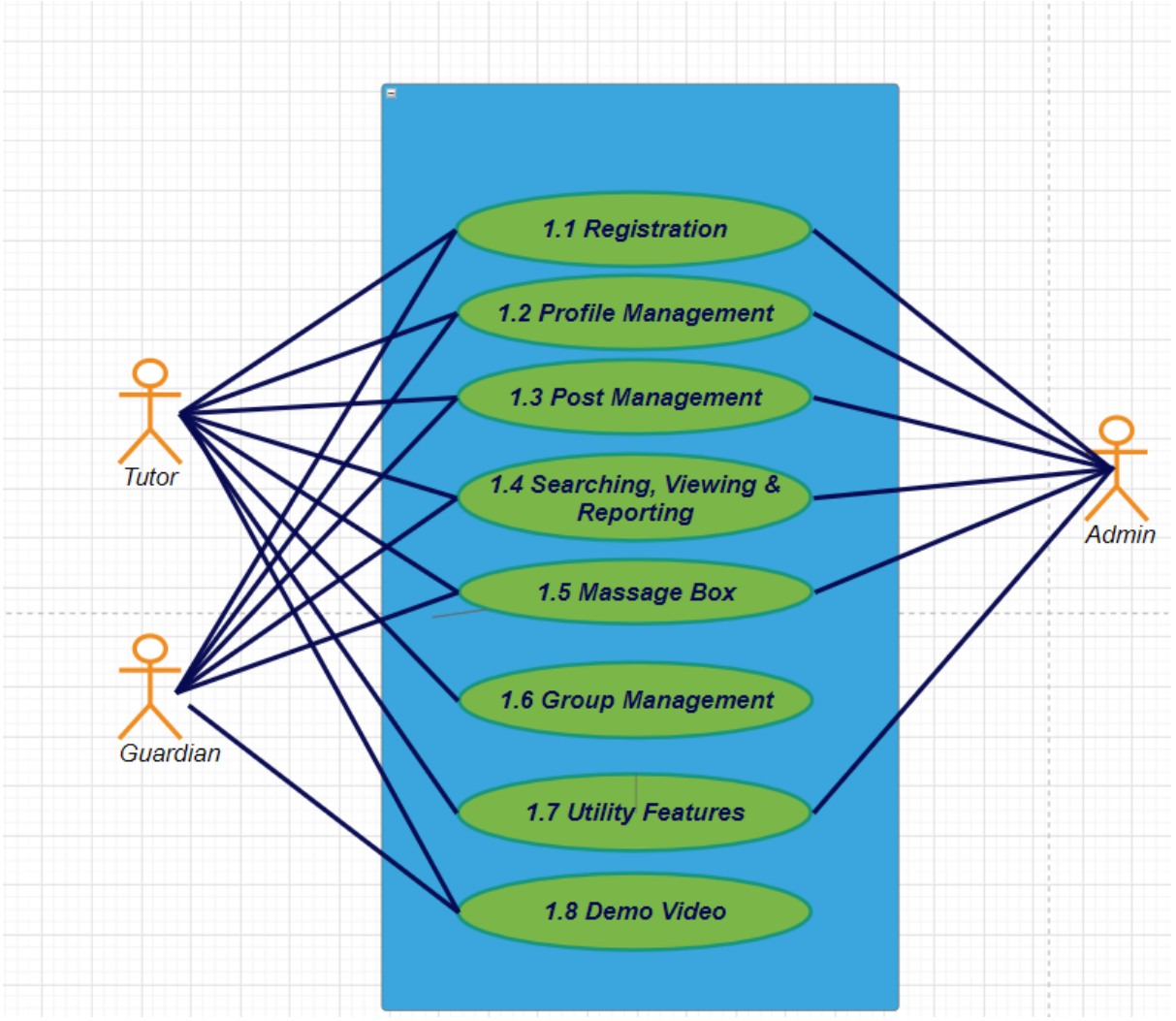


Figure 2: Use case diagram-2

Description: From this level all the subsystems of the proposed main system and connectivity of those subsystems through actors has been explicit. From this level interaction between actors and subsystems will be clearer. Here, the whole system is divided into seven subsystems and E-mail & SMS are the outside system in this proposed system.

3. Use case diagram level: 1.1

ID: L-1.1
Name: Registration
Primary Actor: Tutor, Guardian, Admin

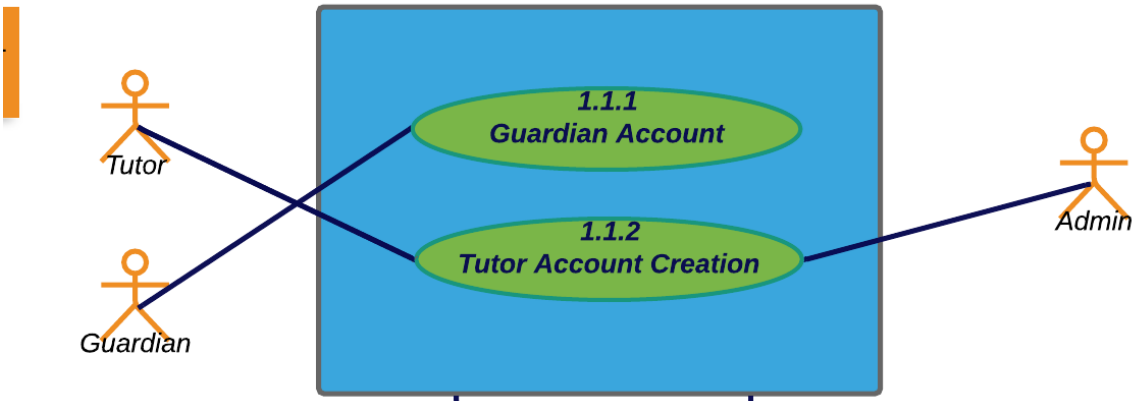


Figure 3: Use case diagram-3

Description : The registration system is divided into two more subsystems. They are Guardian Account Creation and Tutor Account Creation. Here, E-mail & SMS are the outside system in this proposed system.

4. Use case diagram level: 1.1.1

ID: L-1.1.1
Name: Guardian Account Creation
Primary Actor: Guardian
Secondary Actor: SMS

Figure 4: Use case diagram-4

5. Use case diagram level: 1.1.2

ID: L-1.1.2
Name: Tutor Account Creation
Primary Actor: Tutor
Secondary Actor:

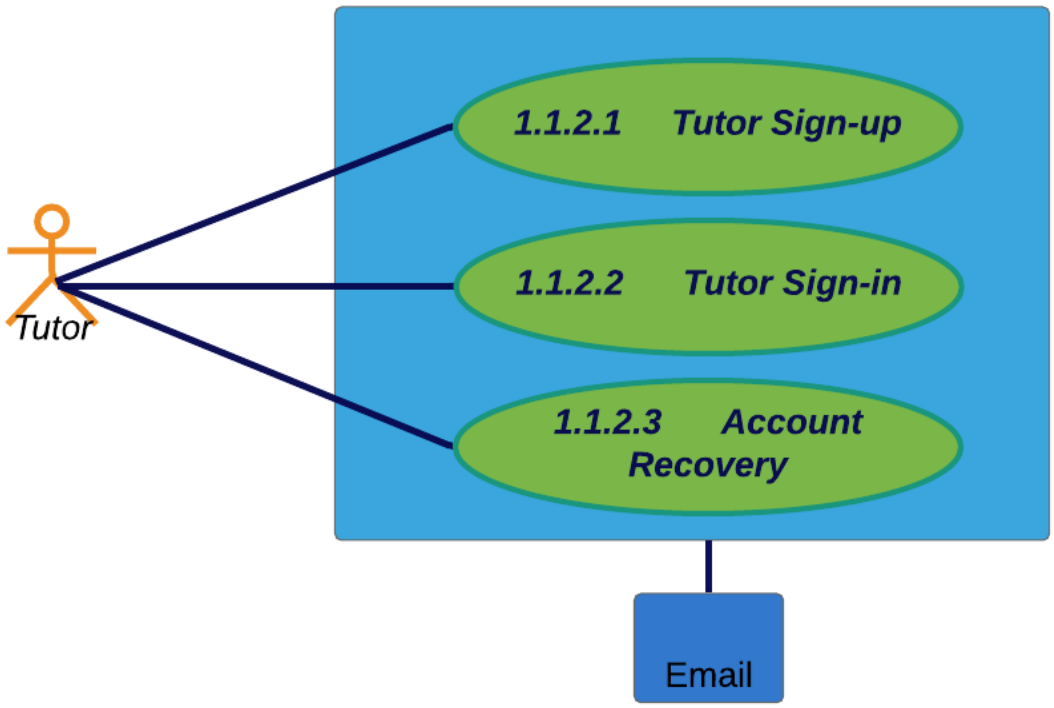


Figure 5: Use case diagram-5

Description : The Tutor Account Creation subsystem is divided into three more subsystems. They are Tutor Sign Up, Sign In, Account Recovery. Here, E-mail & SMS are the outside system in this proposed system.

6. Use case diagram level: 1.2.2.1

ID: L-1.2.2.1
Name: Sign Up
Primary Actor: Tutor, Admin
Secondary Actor: Email, SMS

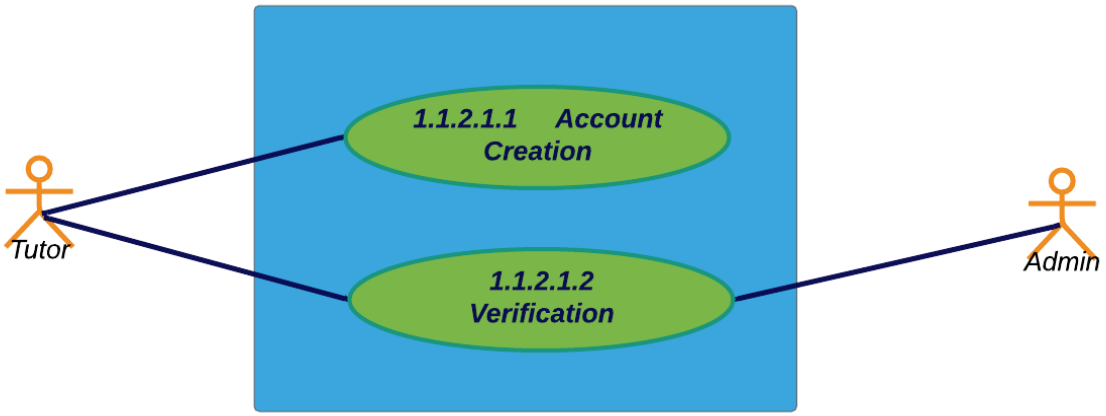


Figure 6: Use case diagram-6

7. Use case diagram level: 1.2

ID: L-1.2
Name: Profile Management
Primary Actor: Tutor, Guardian
Secondary Actor:

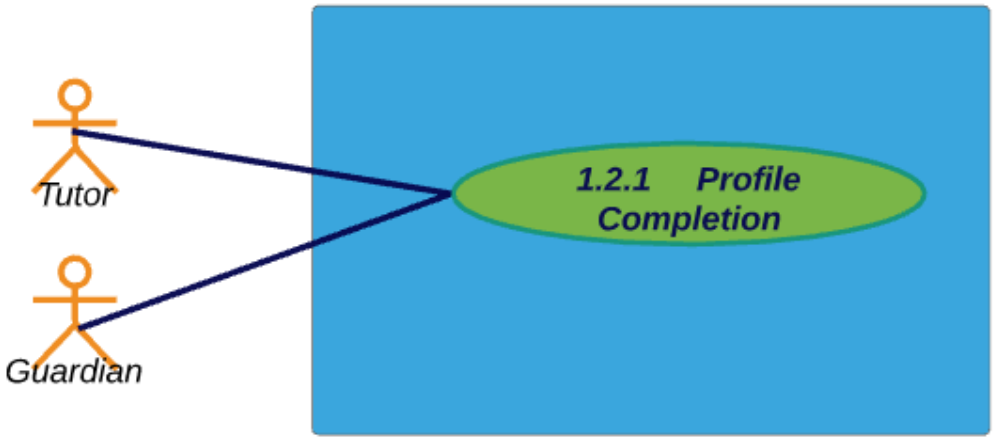


Figure 7: Use case diagram-7

8. Use case diagram level: 1.2.1

ID: L-1.2.1
Name: Profile Completion
Primary Actor: Tutor, Guardian, Admin
Secondary Actor:

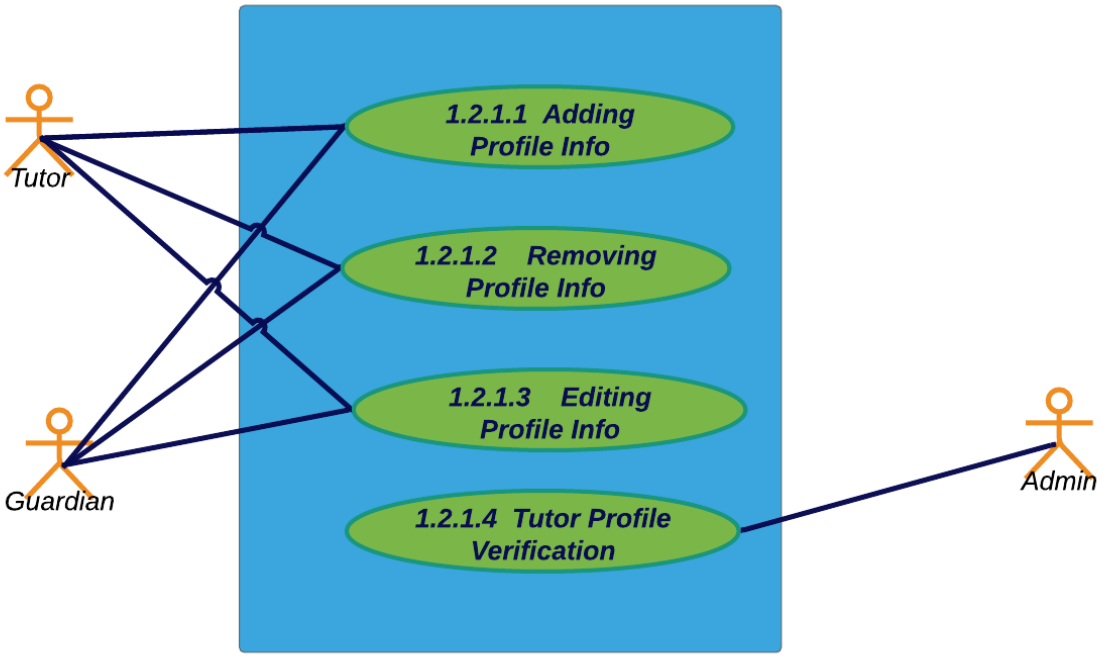


Figure 8: Use case diagram-8

Description: Divided into four subsystems : AddingProfileInfo, RemovingProfileInfo, EditingProfileInfo.

10. Use case diagram level: 1.3

ID: L-1.3
Name: Post Management
Primary Actor: Tutor, Guardian

Description : In this subsystem, Guardian can create post, edit post and remove it. Guardian also can set up availability of the post. A tutor can view the posts .

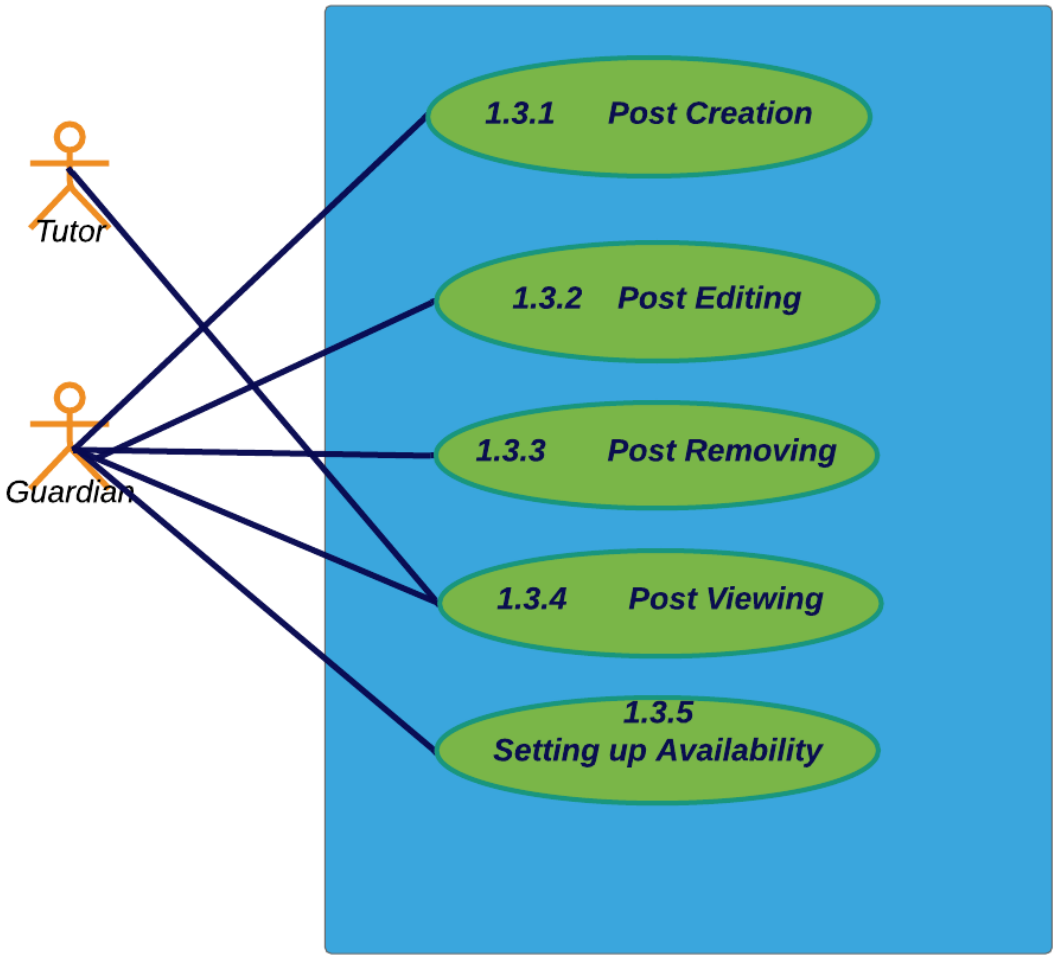


Figure 10: Use case diagram-10

11. Use case diagram level: 1.4

ID: L-1.4
Name: Searching, Viewing and Reporting
Primary Actor: Tutor, Guardian, Admin
Secondary Actor:

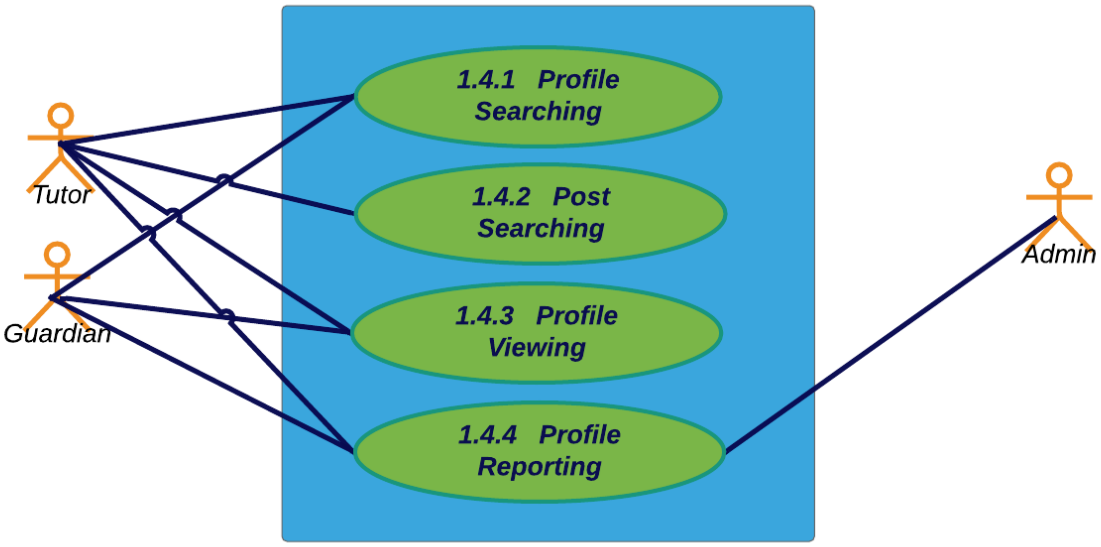


Figure 11: Use case diagram-11

12. Use case diagram level: 1.5

ID: L-1.5
Name: MessageBox
Primary Actor: Tutor, Guardian
Secondary Actor:

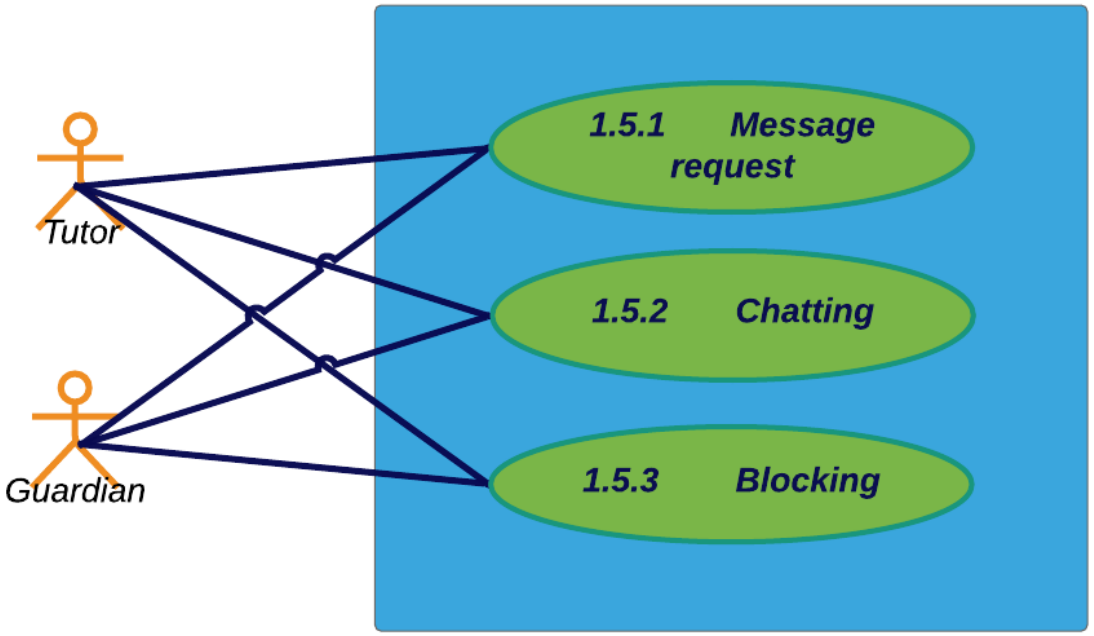


Figure 12: Use case diagram-12

13. Use case diagram level: 1.6

ID: L-1.6
Name: Group Management
Primary Actor: Group Admin, Tutor
Secondary Actor:

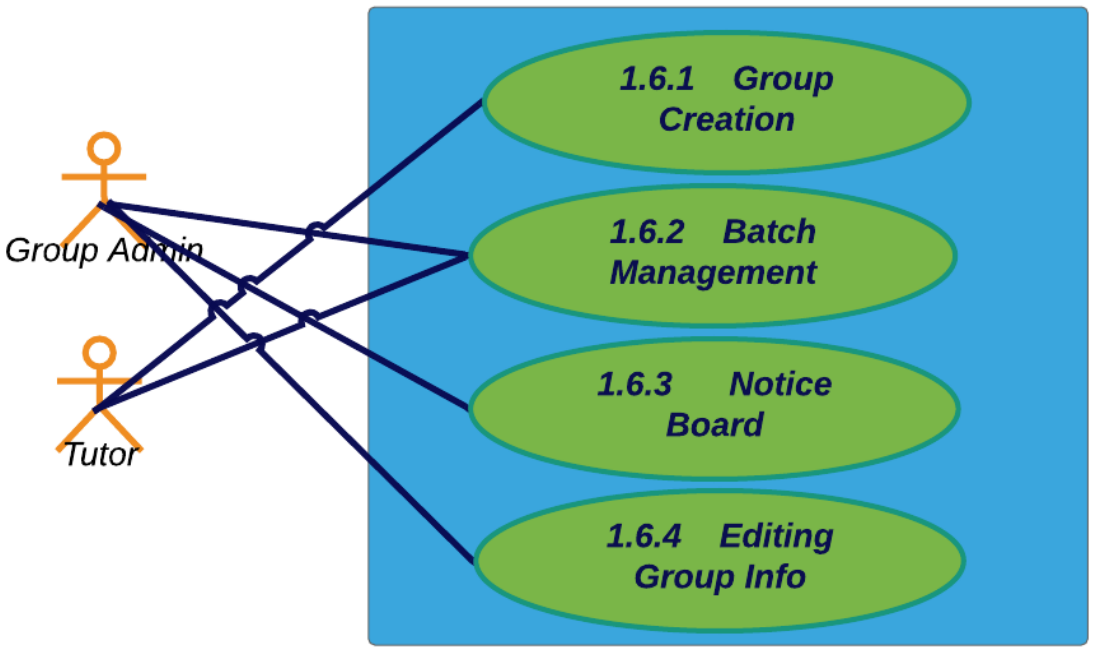


Figure 13: Use case diagram-13

14. Use case diagram level: 1.6.1

ID: L-1.6.1
Name: Batch Management
Primary Actor: Group Admin, Tutor
Secondary Actor:

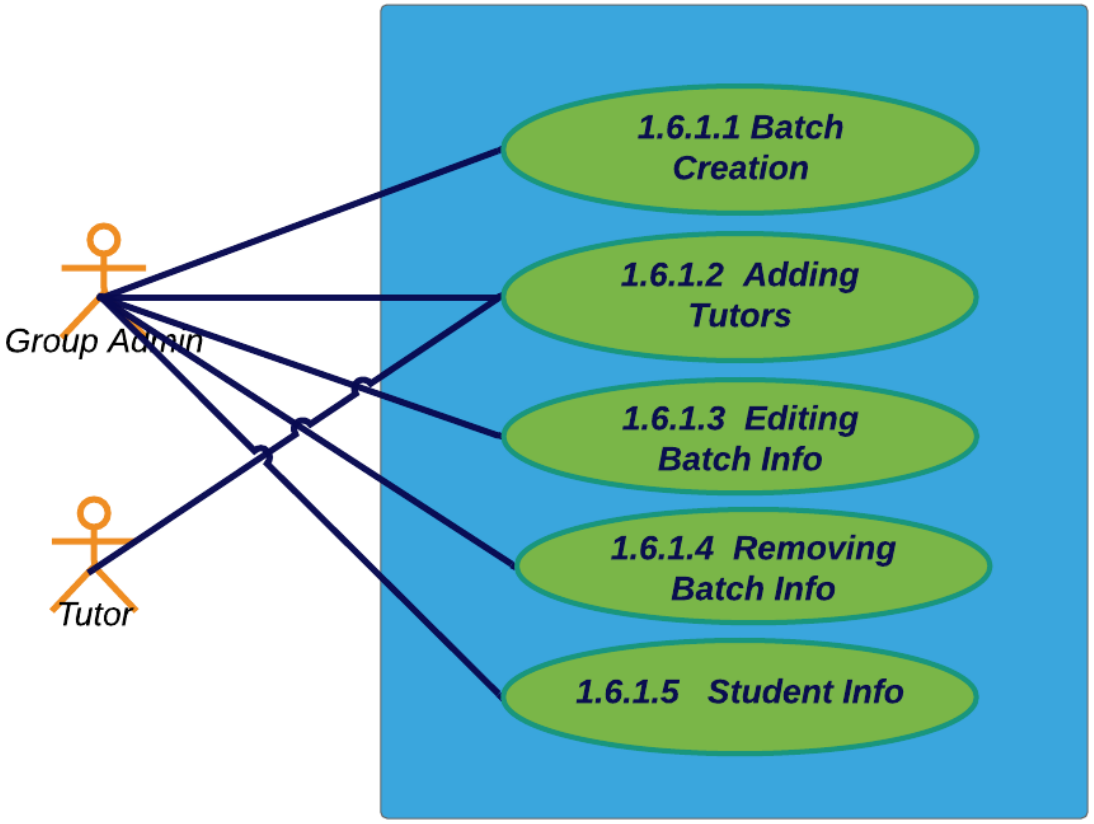


Figure 14: Use case diagram-14

15. Use case diagram level: 1.7

ID: L-1.7
Name: Utility Features
Primary Actor: Tutor
Secondary Actor:

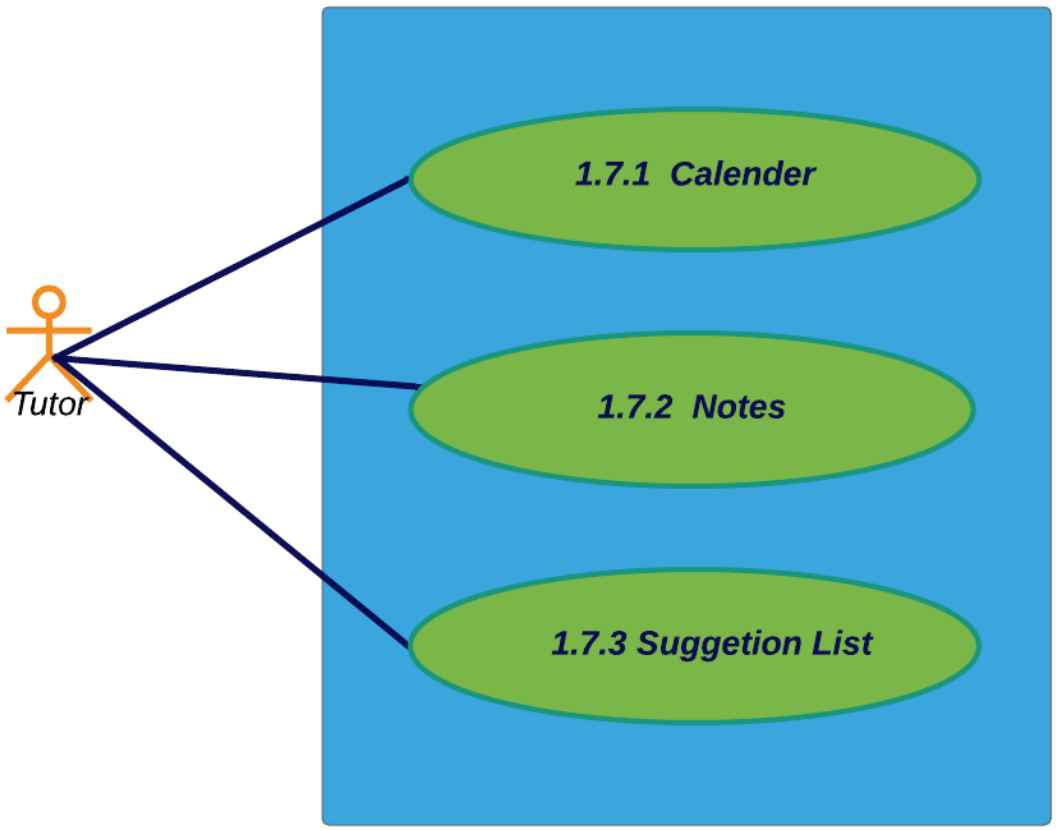


Figure 15: Use case diagram-15

15. Use case diagram level: 1.8.1

ID: L-1.8.1
Name: Demo Video
Primary Actor: Tutor, Guardian
Secondary Actor:

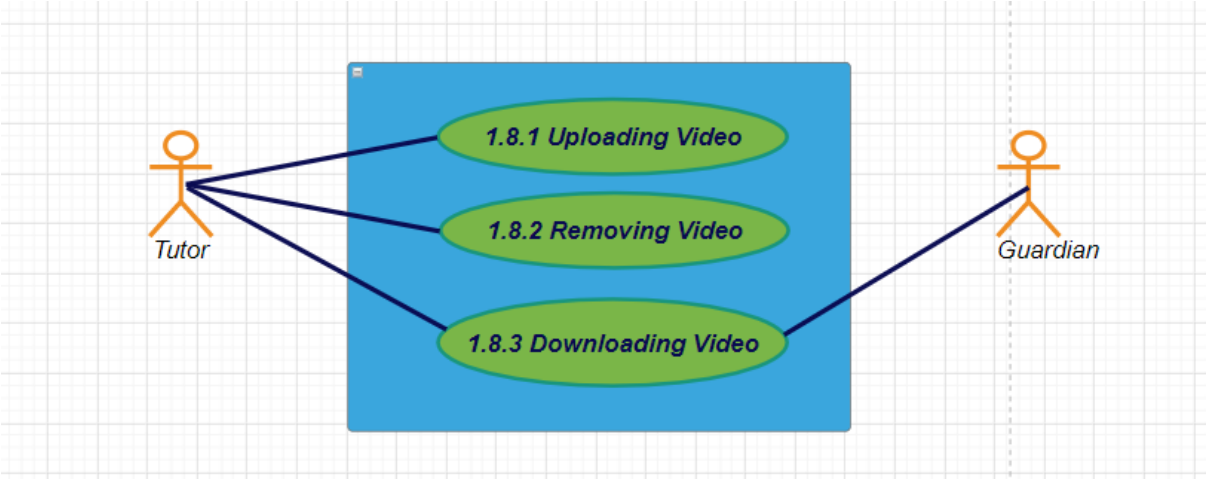


Figure 16: Use case diagram-16

4.4 Activity Diagrams

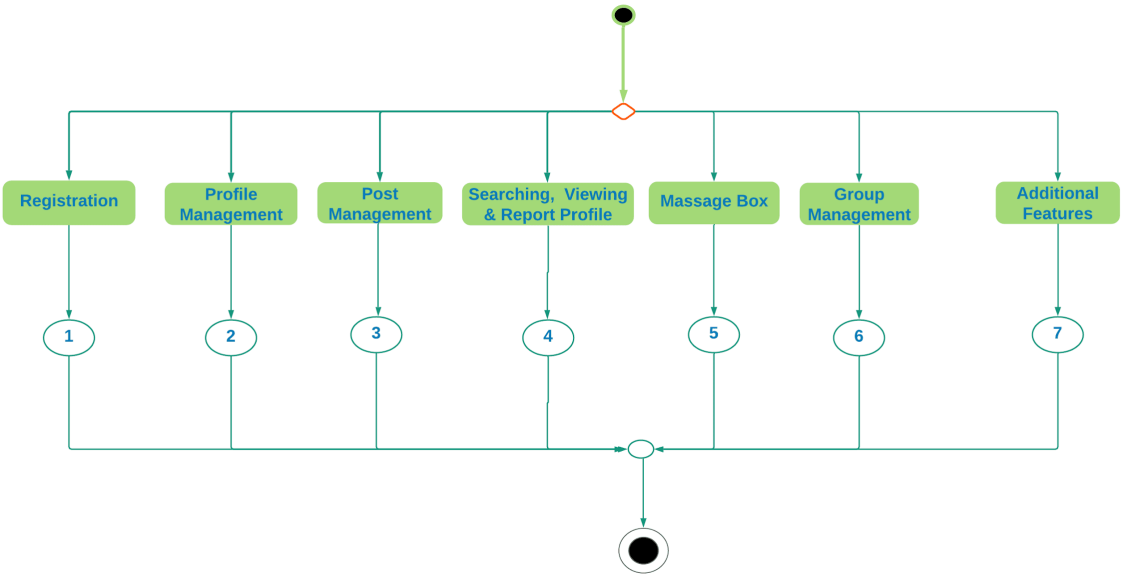


Figure 16: Activity diagram-1

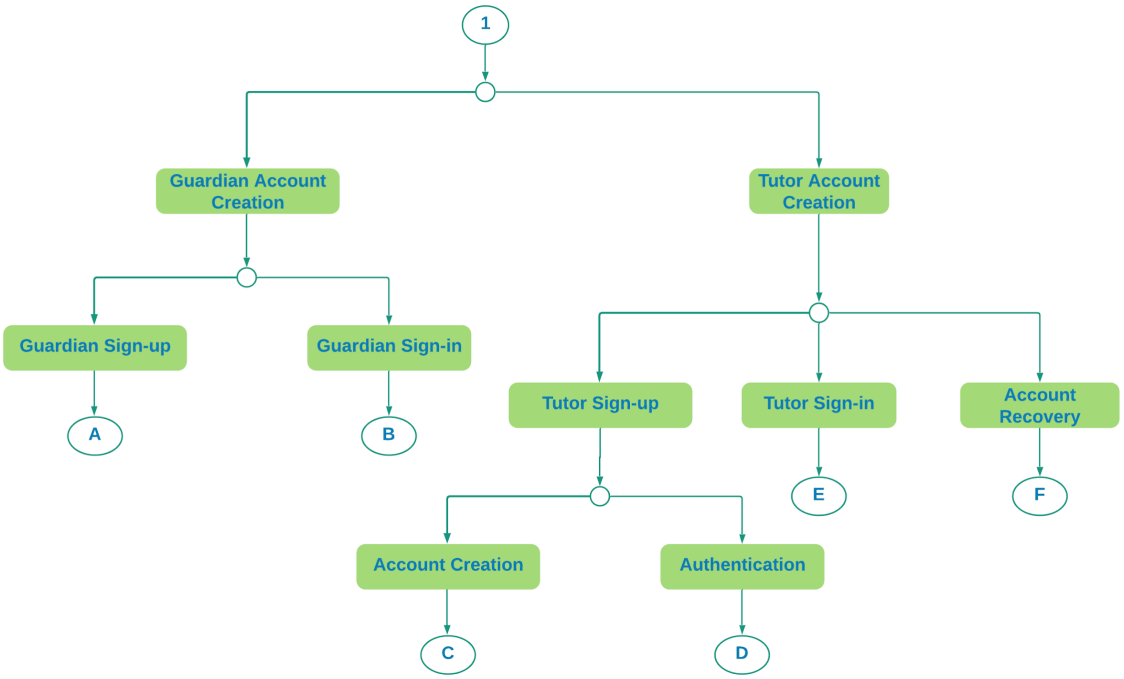


Figure 17: Activity diagram-2

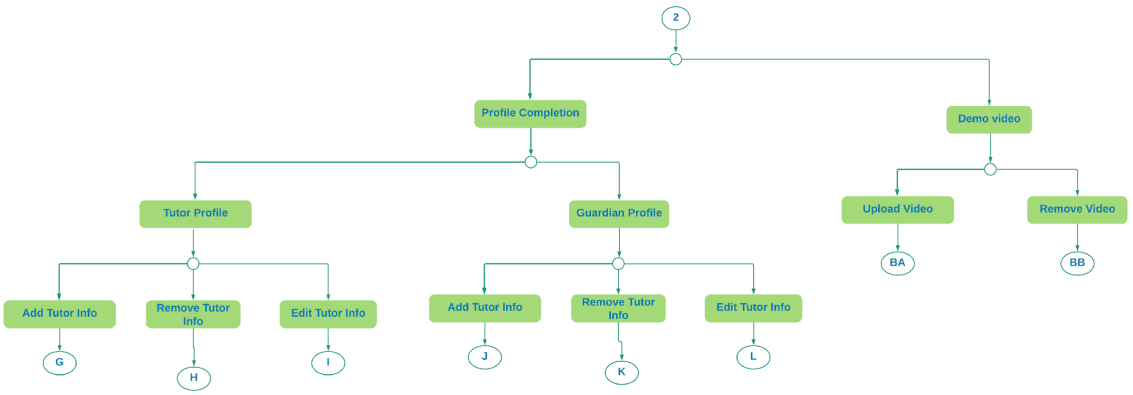


Figure 18: Activity diagram-3

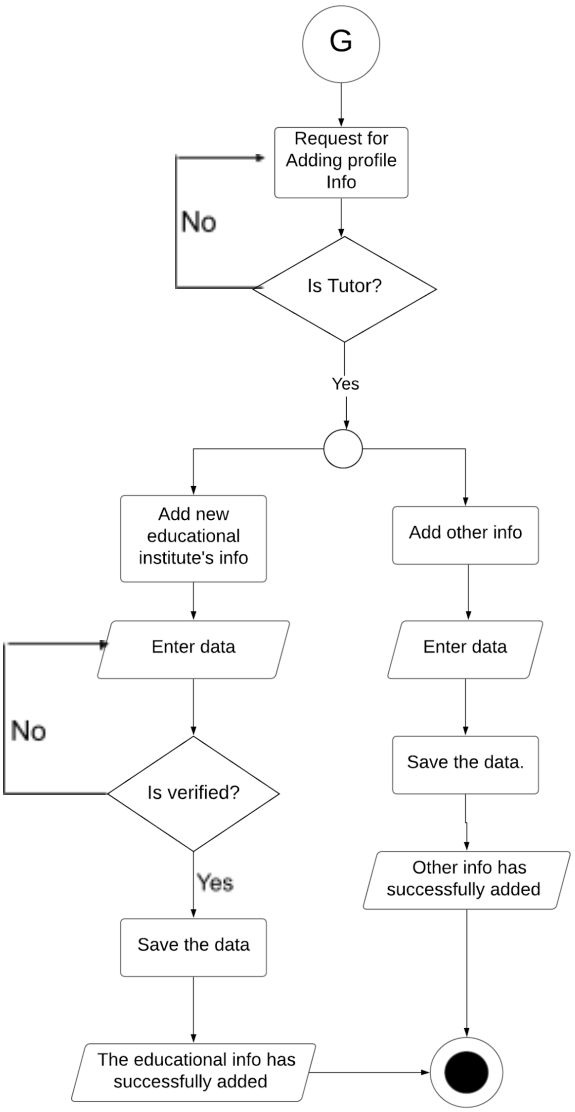


Figure 19: Activity diagram-4

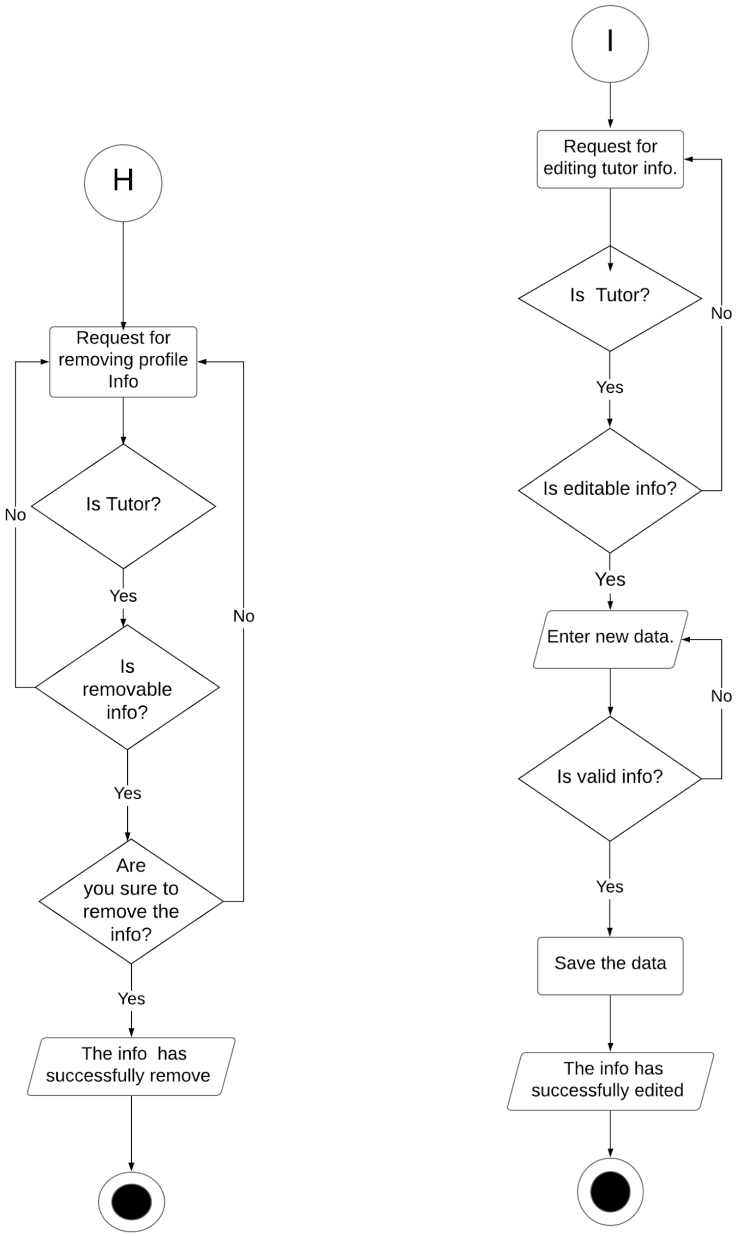


Figure 20, 21: Activity diagram-5,6

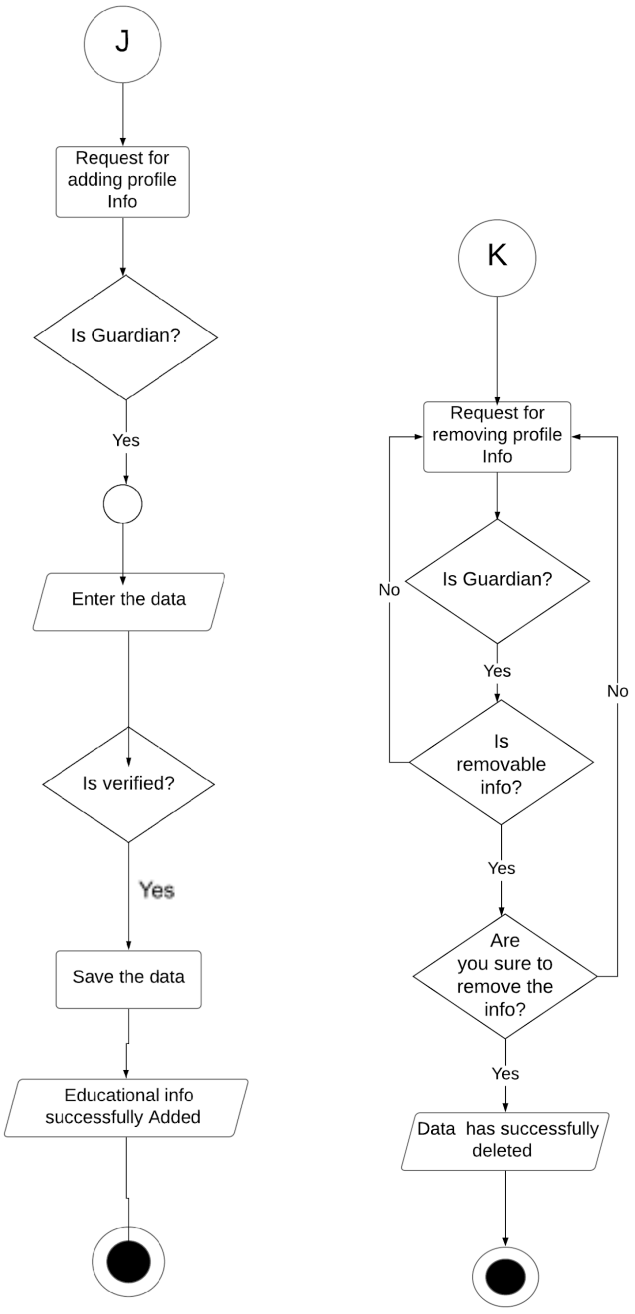


Figure 22, 23: Activity diagram-7,8

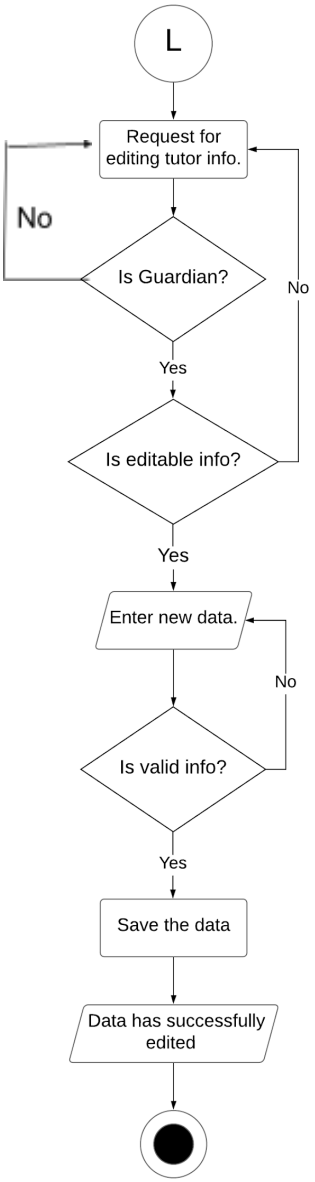


Figure 24: Activity diagram-9

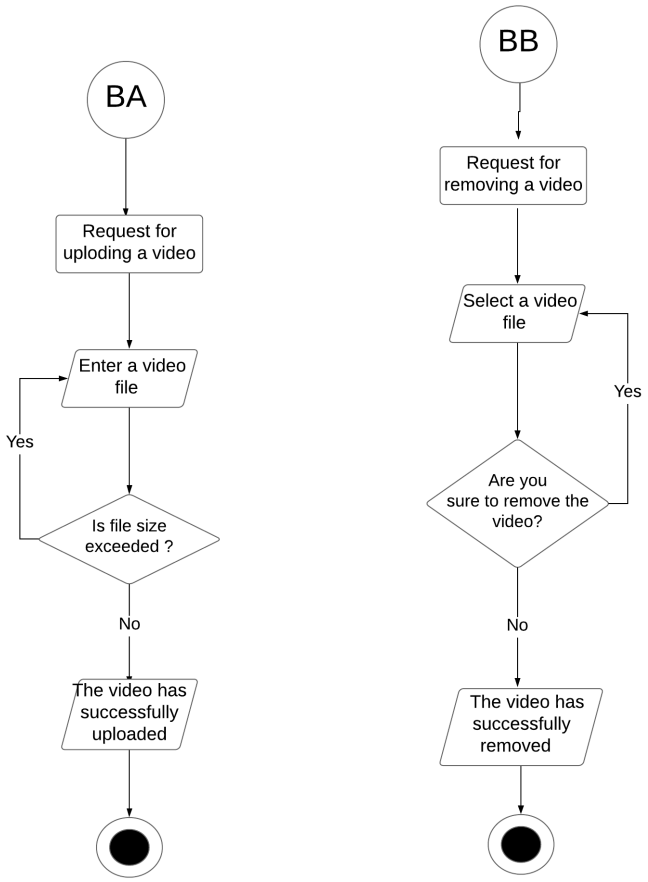


Figure 25, 26: Activity diagram-10, 11

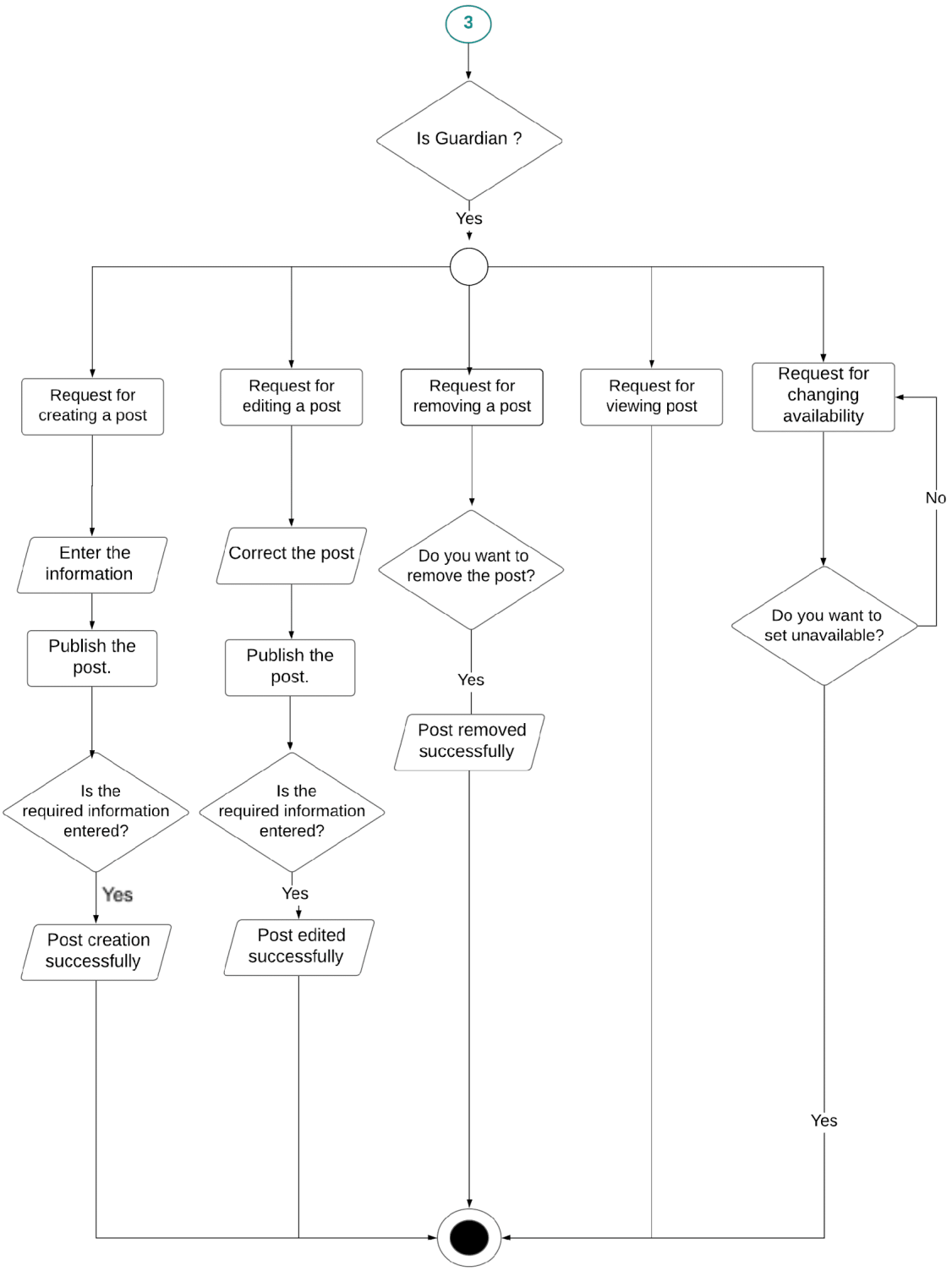


Figure 27: Activity diagram-12

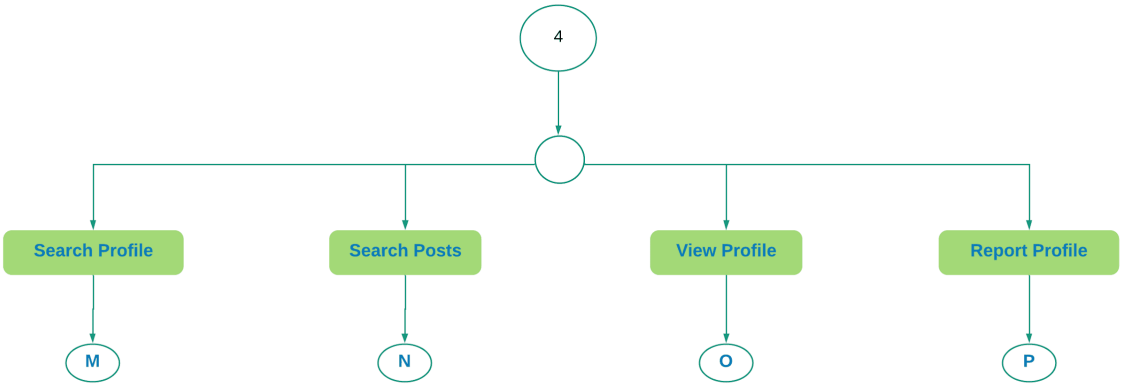


Figure 28: Activity diagram-13

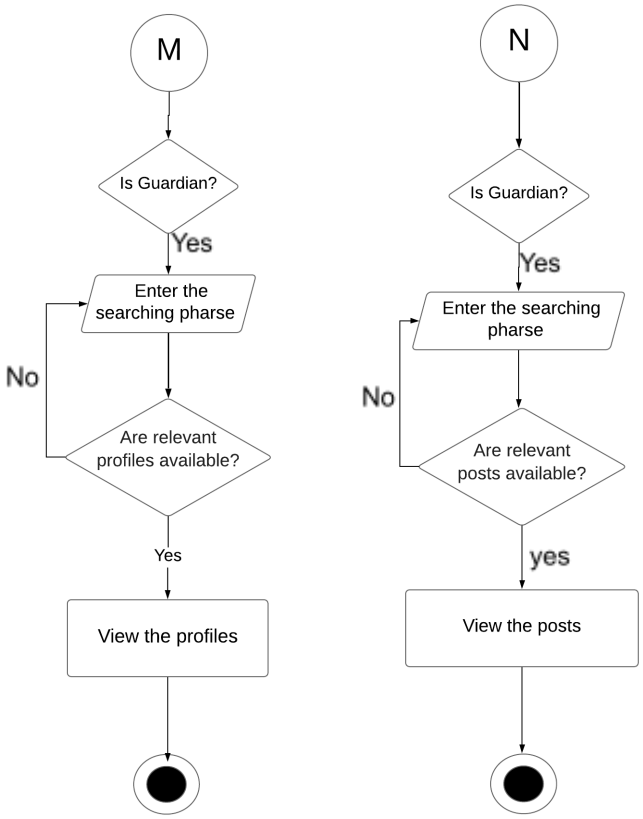


Figure 29, 30: Activity diagram-14, 15

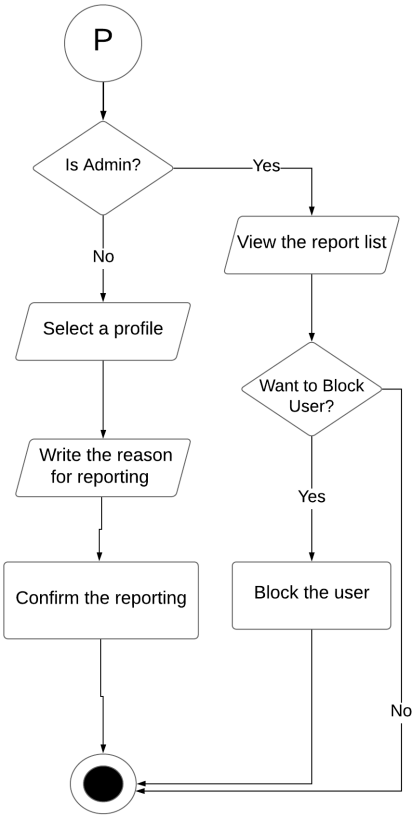


Figure 31: Activity diagram-16

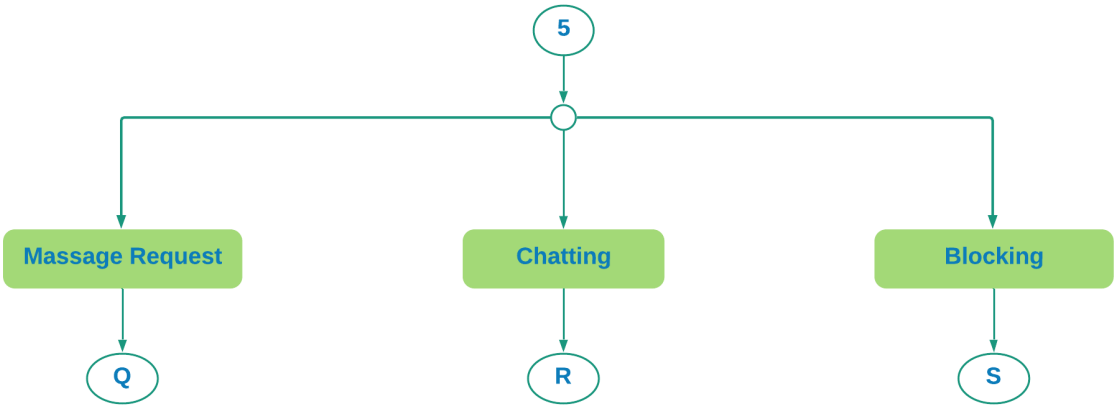


Figure 32: Activity diagram-17

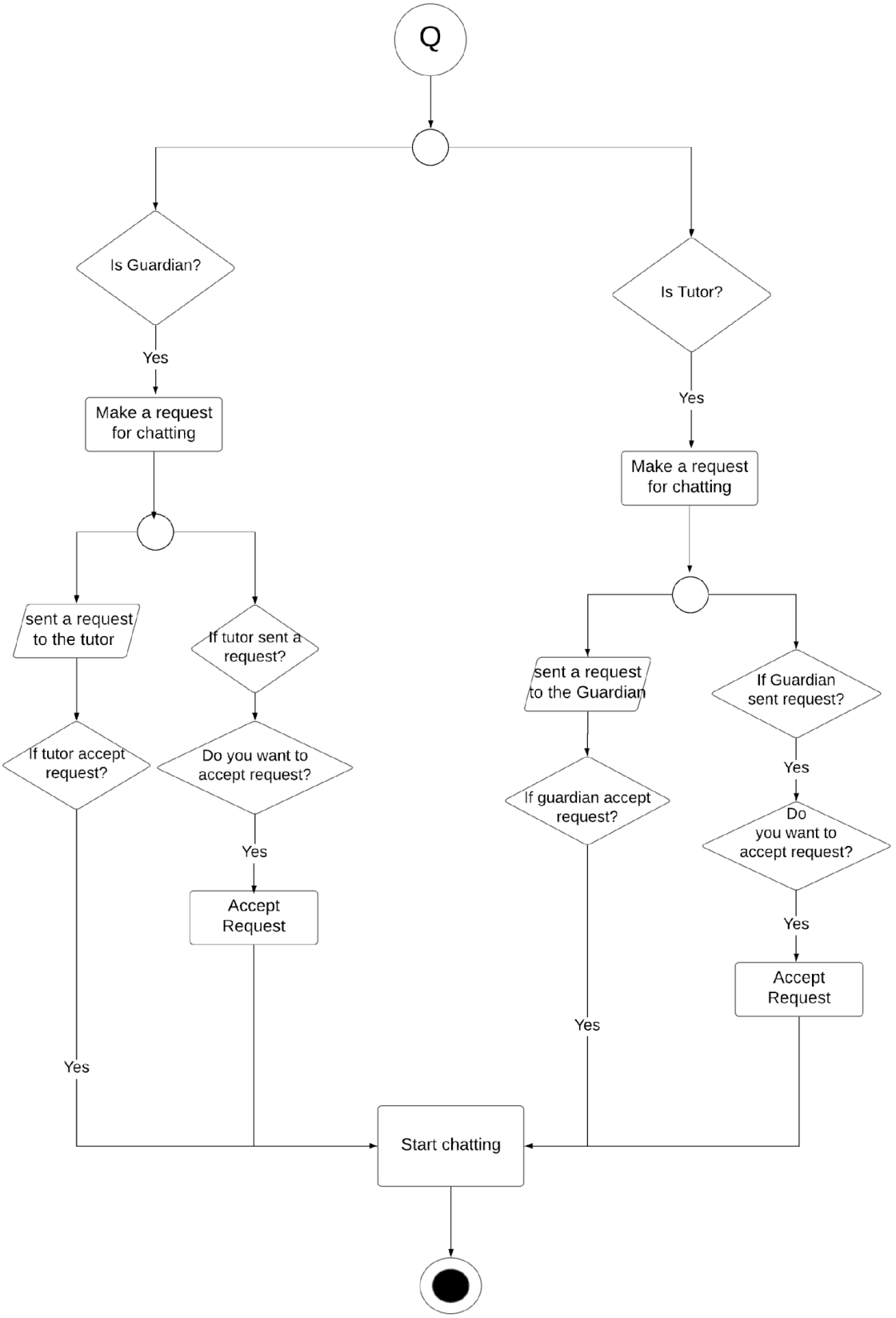


Figure 33: Activity diagram-18

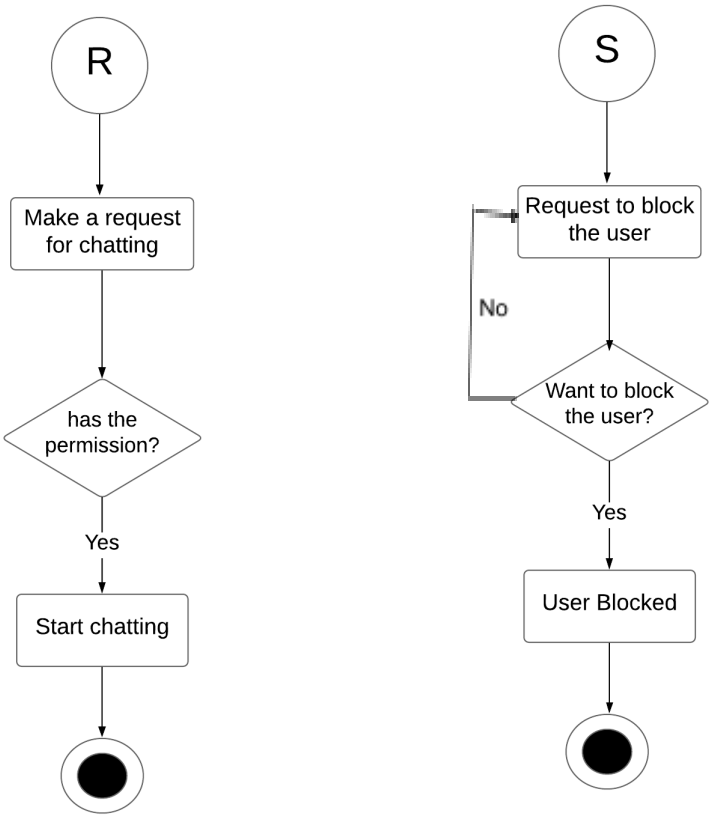


Figure 34: Activity diagram-19

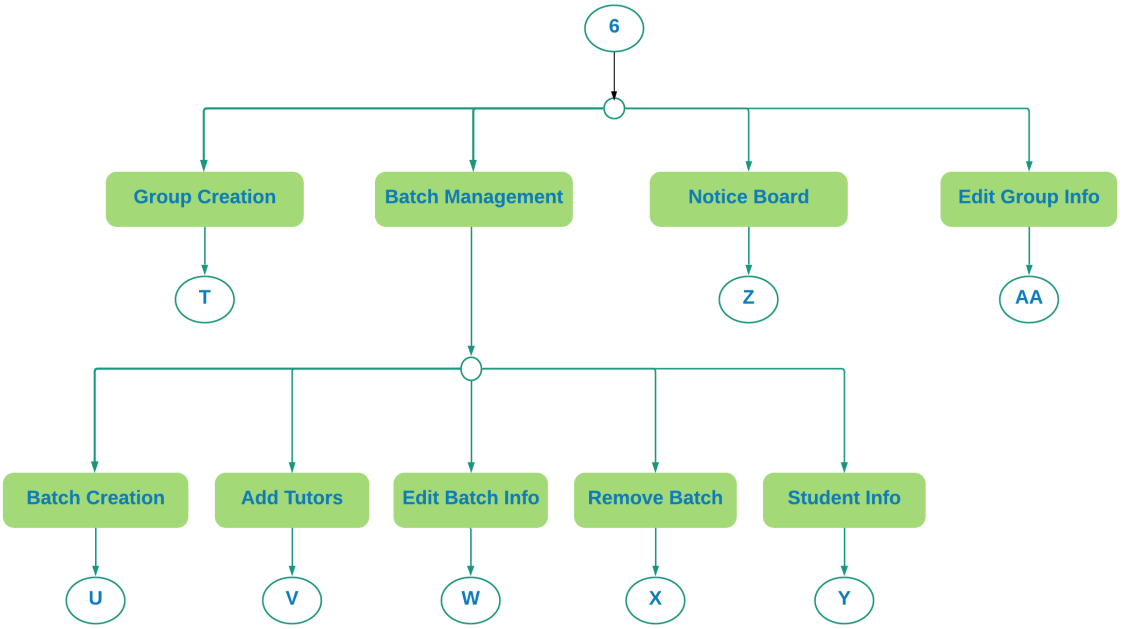
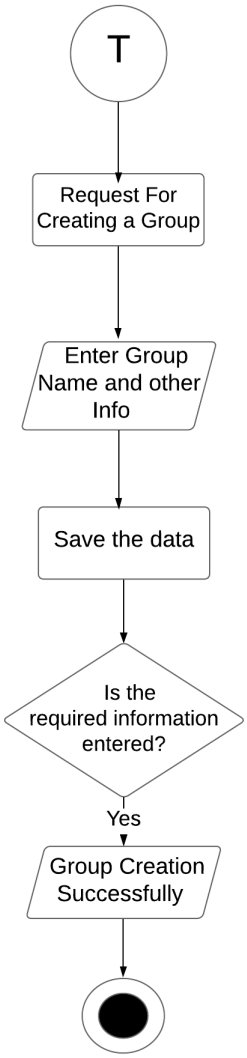


Figure 35: Activity diagram-20



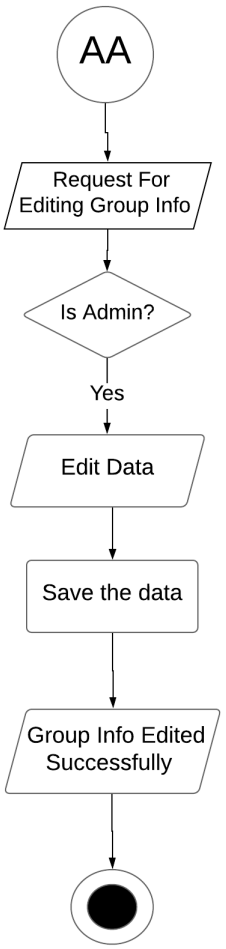


Figure 36, 37: Activity diagram-21, 22

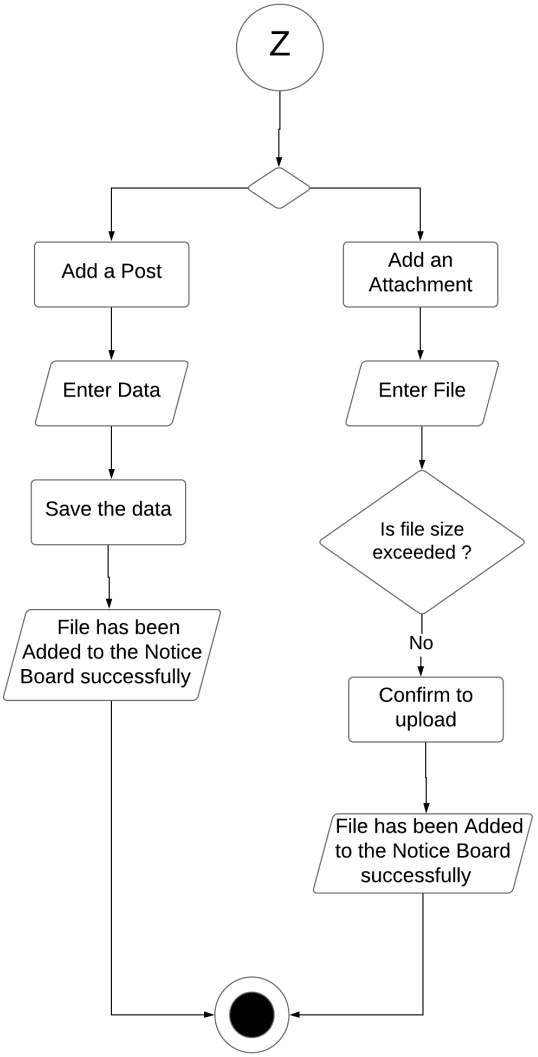


Figure 38: Activity diagram-23

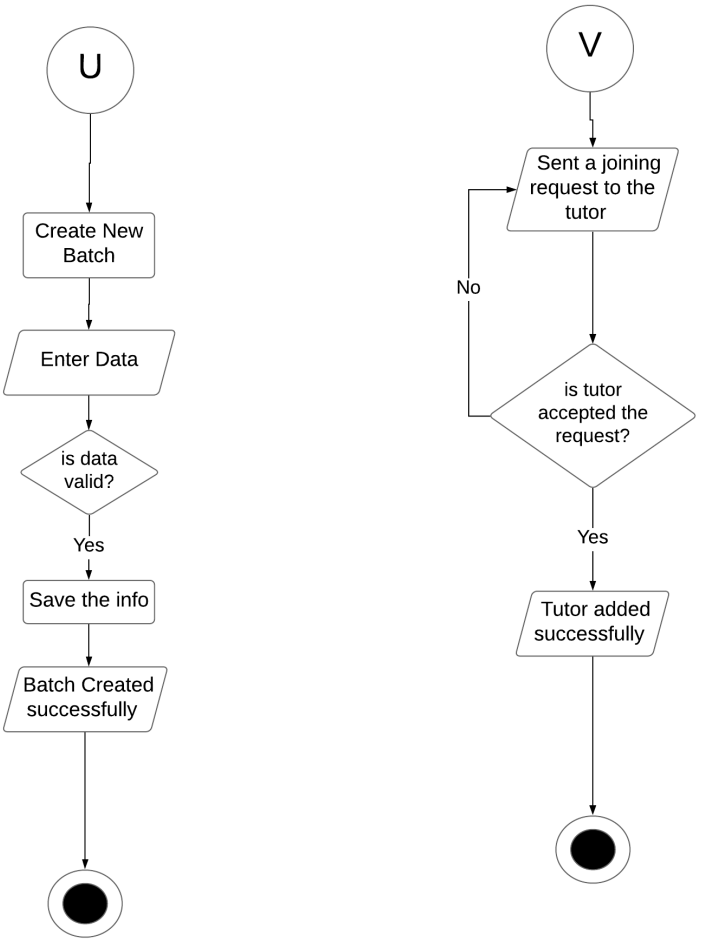


Figure 39, 40: Activity diagram-24, 25

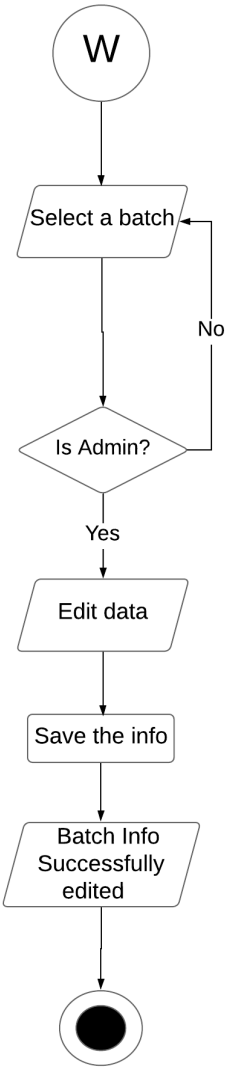


Figure 41: Activity diagram-26

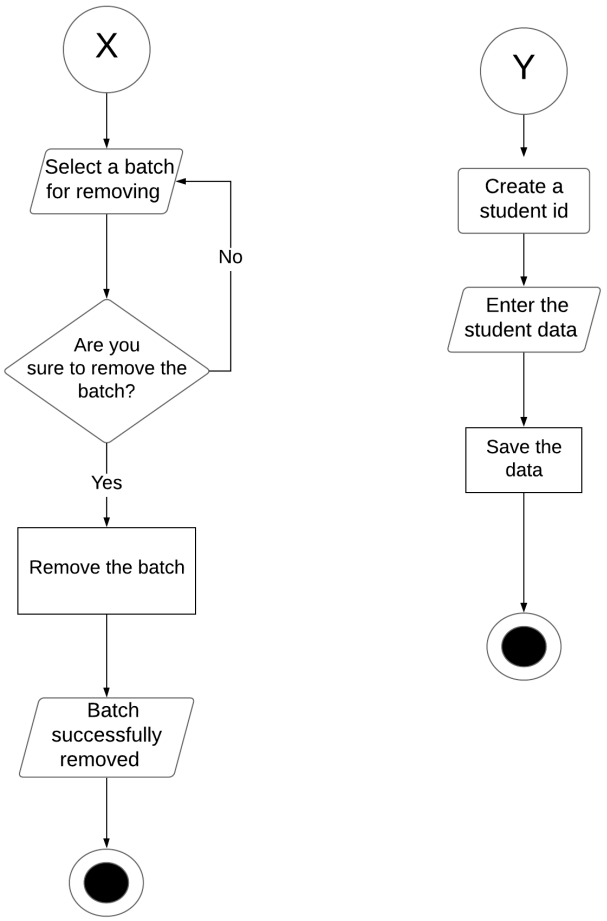


Figure 42, 43: Activity diagram-27, 28

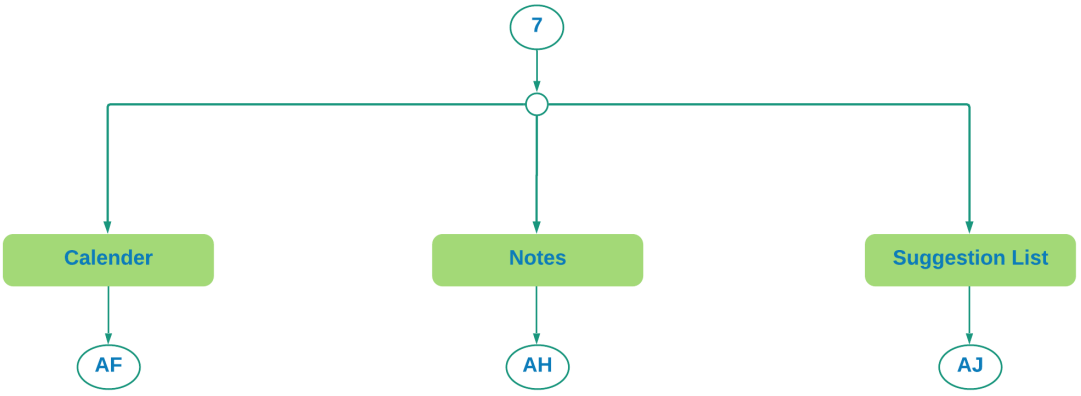


Figure 44: Activity diagram-29

4.5 Swimlane

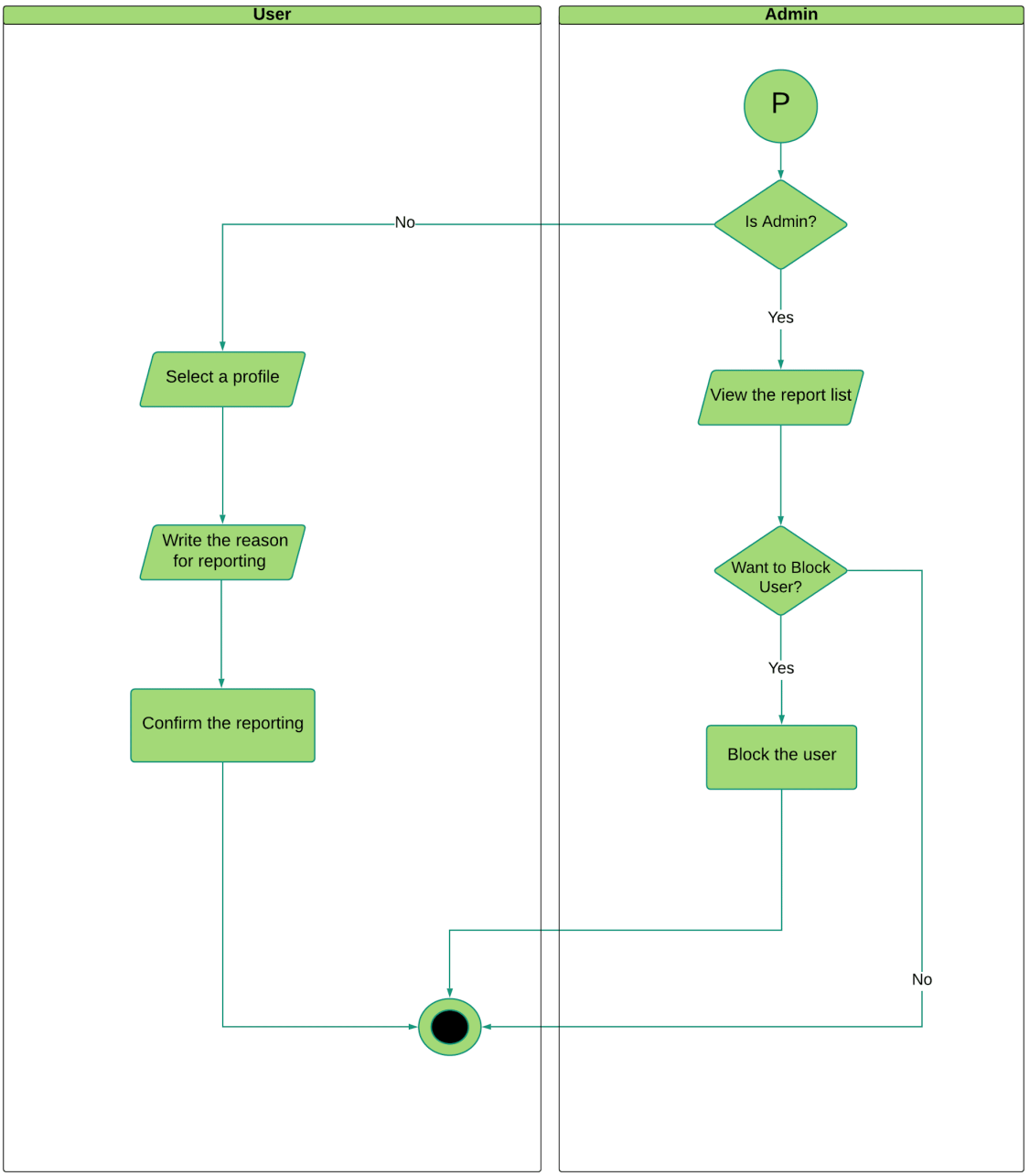


Figure 45: Swim lane diagram-30

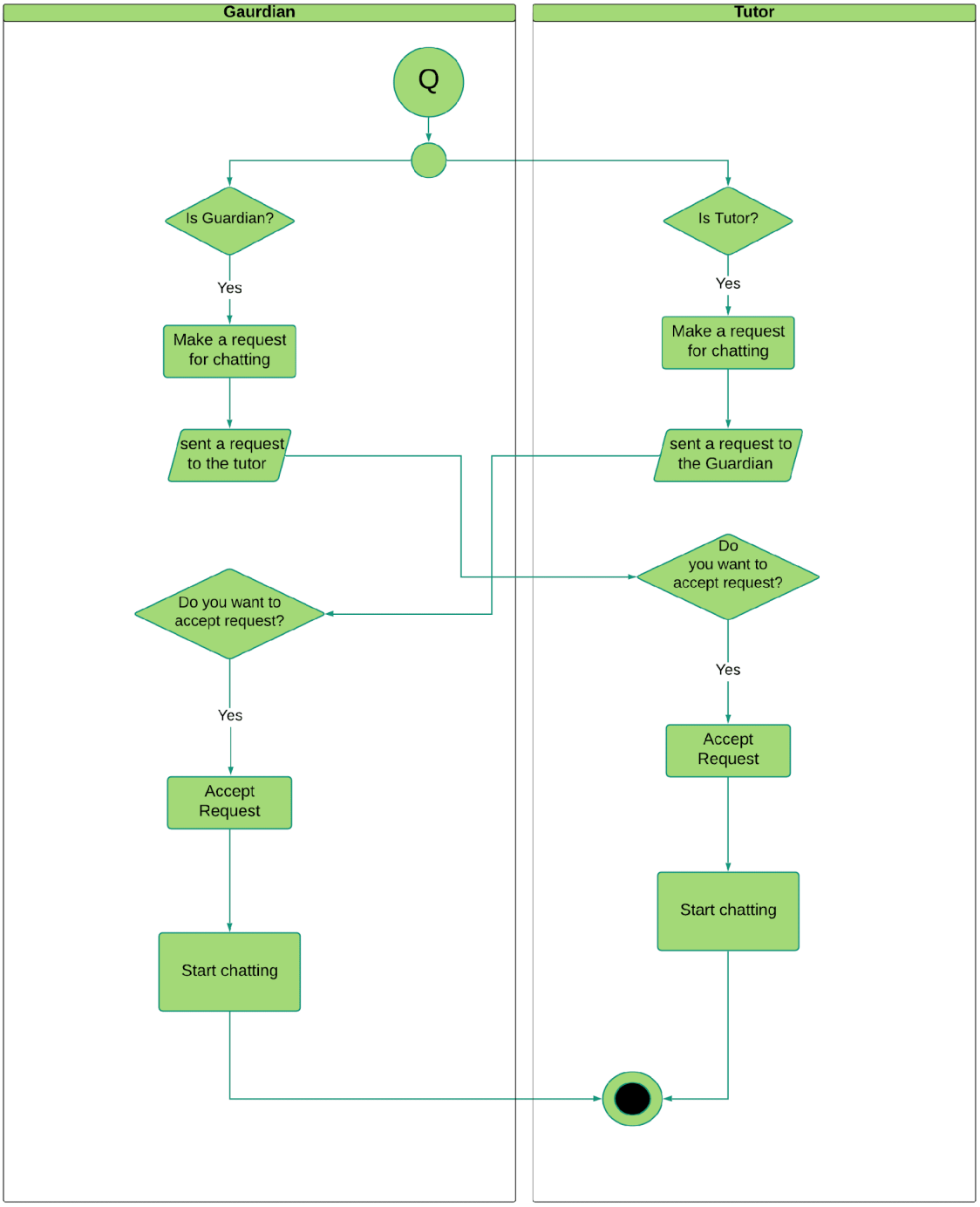


Figure 45: Swim lane diagram-30

CHAPTER 5

DATA BASED MODELING

5.1 Introduction

Sometimes software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated. The software team chooses to create data models as a part of overall requirements modelling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

5.2 Data objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

5.2.1 Noun identification

We identified all the nouns whether they are in problem space or in solution space from our usage scenario.

Table 1: Noun Identification for Data Modeling

Serial	Noun	S/P	Attributes
1	Project	P	x
2	Tuition	P	x
3	Guardian	S	6, 11, 12, 13
4	Student	S	6, 11, 12,13

5	Tutor	**S**	9-19
6	Username	**S**	x
7	System	P	x
8	Account	P	x
9	First Name	**S**	x
10	Last Name	**S**	x
11	Address	**S**	x
12	Mobile Number	**S**	x
13	Email	**S**	x
14	Gender	**S**	x
15	Edu. Institute's name	**S**	x
16	Tutor's Subject	**S**	x
17	Current Position	**S**	x
18	Identity Card picture	**S**	x
19	Reference	**S**	x
20	Verification Code	P	x
21	Password	**S**	x
22	Verified Tutor	**S**	9-19, 29-36
23	Authentication	P	x
24	Admin	**S**	13, 21
25	Authenticated Tutor	P	x
26	Fake Information	P	x
27	Profile	P	x
28	Default Picture	P	x
29	Profile Picture	**S**	x
30	Medium of Version	**S**	x
31	Preferred Classes	**S**	x
32	Preferred Group	**S**	x
33	Preferred Subject	**S**	x
34	Preferred Area	**S**	x
35	Experience Status	**S**	x
36	Minimum Salary	**S**	x
37	Notification	P	x

38	User	P	x
39	Date Id	S	x
40	Reminder Id	S	x
41	Note Id	**S**	--
42	Calendar	**S**	39,40
43	Note	**S**	41
44	Demo Video	**S**	--
45	Tuition Post	**S**	46-59
46	Title of Post	**S**	x
47	Student Institute's name	**S**	x
48	Medium/Version	**S**	x
49	Student Class	**S**	x
50	Student Group	**S**	x
51	Subject List	**S**	x
52	Tutor Gender Preference	**S**	x
53	Days per Week	**S**	x
54	Student Area Address	**S**	x
55	Details Address	**S**	x
56	Contact Number	**S**	x
57	Salary Range	**S**	x
58	Others Option	**S**	x
59	Availability	**S**	x
60	Message Box	P	x
61	Default Notification	P	x
62	Reminder Notification	P	x
63	Availability of Post	P	x
64	Searching	P	x
65	Viewing	P	x
66	Initial Message	P	x
67	Message Request	P	x
68	Group	**S**	69-71
69	Group Name	**S**	x
70	Group Address	**S**	x

71	Group Image	**S**	x
72	Group Admin	**S**	9-19, 29-36
73	Batch	**S**	74-78
74	Batch Name	**S**	x
75	Payment	**S**	x
76	Schedule	**S**	x
77	Seat's Availability	**S**	x
78	Academic Days	**S**	x
79	Batch Info	P	x
80	Student Info	**S**	81-85
81	Student Name	**S**	x
82	Student Address	**S**	x
83	Student Phone Number	**S**	x
84	Student's Institute's Name	**S**	x
85	Student Class Position	**S**	x
86	Notice Board	P	x
87	Notice	**S**	88, 89
88	Post	**S**	x
89	Attachment	**S**	x
90	Files	**P**	x
91	Blocking Option	**P**	x

5.2.2 Selected Initial Data Objects:

Serial	Noun	S/P	Attributes
1	Guardian	**S**	6, 11, 12, 13
2	Student	**S**	6, 11, 12, 13
3	Tutor	**S**	9-19
4	Verified Tutor	**S**	9-19, 29-36
5	Admin	**S**	13, 21

6	Tuition Post	**S**	46-59
7	Group	**S**	69-71
8	Group Admin	**S**	9-19, 29-36
9	Batch	**S**	74-78
10	Student Info	**S**	81-85
11	Notice	**S**	88, 89
12	Calendar	**S**	39,40
13	Note	**S**	41

5.2.3 Analysis:

Here, Guardian and Student are identical. So, we are going to merge them and create Guardian.

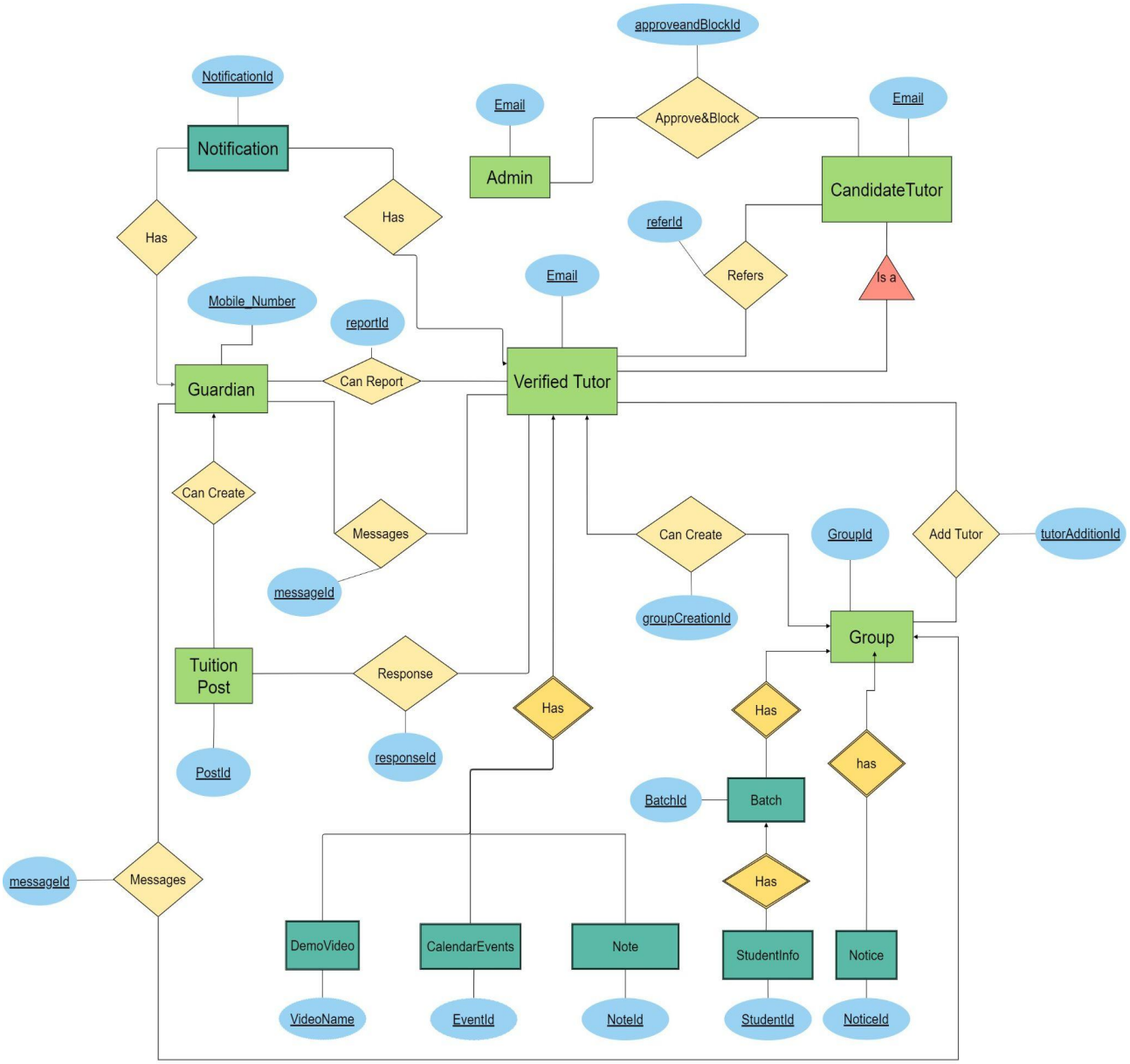
5.2.4 Final Data Objects

Serial	Noun	Attributes
1	Guardian	name address mobileNumber profilePicture
2	CandidateTutor	name address mobileNumber email gender institutionName department currentPosition identityCardPicture

3	VerifiedTutor	firstName lastName address mobileNumber email gender institutionName subject currentPosition identityCardPicture reference profilePicture preferredMedium/Version preferredClasses preferredGroup preferredSubjects preferredAreas experienceStatus minimumSalary
4	Admin	email password
5	Post	TitleOfPost StudentInstituteName Medium/Version StudentClass Group SubjectList TutorGenderPreference DaysPerWeek StudentAreaAddress DetailAddress ContactNumber SalaryRange OthersOption Availability
6	Group	GroupName GroupAddress GroupImage
7	Group Admin	firstName lastName address mobileNumber email gender institutionName subject currentPosition

		identityCardPicture reference profilePicture preferredMedium/Version preferredClasses preferredGroup preferredSubjects preferredAreas experienceStatus minimumSalary
8	Batch	BatchName Payment Schedule SeatAvailability AcademicDays
9	StudentInfo	StudentName StudentAddress StudentPhoneNumber StudentInstituteName StudentClassPosition
10	Notice	Post Attachment
11	Calendar	eventTitle eventLocation eventDescription eventDate startingTime endingTime attendeesId
12	Note	noteId noteText

5.3 Entity Relation Diagram



5.5 SCHEMA DIAGRAM

GUARDIAN		
Attributes	**Type**	**Size**
username	VARCHAR	100
address	VARCHAR	200
Mobile Number	VARCHAR	15

CANDIDATE TUTOR		
Attributes	**Type**	**Size**
firstName	VARCHAR	20
lastName	VARCHAR	20
address	VARCHAR	200
mobileNumber	VARCHAR	15
email	VARCHAR	50
gender	VARCHAR	10
institutionName	VARCHAR	50
subject	VARCHAR	50
currentPosition	VARCHAR	50
identityCardPicture	BLOB(Binary Large Object)	500KB
isAvailable	BOOLEAN	5

VERIFIED TUTOR		
Attributes	**Type**	**Size**
Tutor.email	VARCHAR	40
profilePicture	BLOB(Binary Large Object)	40
preferredMedium/Version	VARCHAR	20
preferredClasses	VARCHAR	200
preferredGroup	VARCHAR	30
preferredSubjects	VARCHAR	200
preferredAreas	VARCHAR	200
experienceStatus	VARCHAR	50
minimumSalary	VARCHAR	6
DemoVideo	VARCHAR	25MB

ADMIN		
Attributes	**Type**	**Size**
email	VARCHAR	40
password	VARCHAR	40

TUITION POST		
Attributes	**Type**	**Size**
PostId	VARCHAR	80
TitleOfPost	VARCHAR	100
StudentInstituteName	VARCHAR	40
Medium/Version	VARCHAR	15
StudentClass	VARCHAR	200
Group	VARCHAR	50
SubjectList	VARCHAR	200
TutorGenderPreference	VARCHAR	20
DaysPerWeek	VARCHAR	50
StudentAreaAddress	VARCHAR	50
DetailAddress	VARCHAR	200
ContactNumber	VARCHAR	15
SalaryRange	VARCHAR	20
OthersOption	VARCHAR	50
Availability	BOOLEAN	5
PostDate	DATE	8 bytes
PostTime	TIME	8 bytes
Guardian.mobileNumber	VARCHAR	15

MESSAGE		
Attributes	**Type**	**Size**
messageBoxId	VARCHAR	80
Guardian.mobileNumber	VARCHAR	15
VerifiedTutor.Tutor.email	VARCHAR	50
messageRequest	BOOLEAN	40

APPROVE AND BLOCK		
Attributes	**Type**	**Size**
Admin.email	VARCHAR	50
Tutor.email	VARCHAR	50
status	BOOLEAN	5
date&time	DATE&TIME	20

RESPONSE		
Attributes	**Type**	**Size**
TuitionPost.PostID	VARCHAR	15
VerifiedTutor.Tutor.email	VARCHAR	50

REPORT		
Attributes	**Type**	**Size**
Guardian.mobileNumber	VARCHAR	15
VerifiedTutor.Tutor.email	VARCHAR	50
report	BOOLEAN	40

REFERS		
Attributes	**Type**	**Size**
Tutor.email	VARCHAR	50
VerifiedTutor.Tutor.email	VARCHAR	50
reference	BOOLEAN	50

ADD_TUTOR		
Attributes	**Type**	**Size**
GroupId	VARCHAR	50
VerifiedTutor.Tutor.email	VARCHAR	50
Added	BOOLEAN	5

GROUP		
Attributes	**Type**	**Size**
GroupId	VARCHAR	50
GroupName	VARCHAR	50
GroupAddress	VARCHAR	200
GroupImage	BLOB(Binary Large Object)	6000Bytes
VerifiedTutor.Tutor.email	VARCHAR	50

BATCH		
Attributes	**Type**	**Size**
BatchId	VARCHAR	50
BatchName	VARCHAR	50
Payment	VARCHAR	20
Schedule	VARCHAR	200
SeatAvailability	VARCHAR	100
AcademicDays	VARCHAR	50
GroupId	VARCHAR	50

STUDENT INFO		
Attributes	**Type**	**Size**
StudentId	VARCHAR	50
StudentName	VARCHAR	40
StudentAddress	VARCHAR	200
StudentPhoneNumber	VARCHAR	15
StudentInstituteName	VARCHAR	50
StudentClassPosition	VARCHAR	10
GroupId	VARCHAR	50
BatchId	VARCHAR	50

NOTICE		
Attributes	**Type**	**Size**
NoticeId	VARCHAR	50
Post	VARCHAR	40
Attachment	File	1024Bytes
GroupId	VARCHAR	40

CALENDAR EVENTS		
Attributes	**Type**	**Size**
EventId	VARCHAR	50
VerifiedTutor.Tutor.email	VARCHAR	40
eventTitle	Date	8 Bytes
eventLocation	VARCHAR	40
eventDescription	VARCHAR	40
eventDate	Date	8 Bytes
startingTime	Time	5 Bytes
endingTime	Time	5 Bytes
attendeesId	VARCHAR	200

NOTES		
Attributes	**Type**	**Size**
NoteId	VARCHAR	50
VerifiedTutor.Tutor.email	VARCHAR	40
NoteName	VARCHAR	40
NoteText	VARCHAR	400

DEMO VIDEO		
Attributes	**Type**	**Size**
VerifiedTutor.Tutor.email	VARCHAR	50
VideoName	VARCHAR	50
VideoURI	VARCHAR	100

NOTIFICATION		
Attributes	**Type**	**Size**
NotificationID	VARCHAR	50
UserID	VARCHAR	50
type	VARCHAR	50
title	VARCHAR	50
body	VARCHAR	100

CHAPTER 6

CLASS BASED MODELING

This chapter describes the Class Based Model for the "Tutors Planet".

6.1 Class based modelling concept

Class-based modeling is a stage of requirements modelling. It uses common concepts of object-oriented programming to craft an impression of an application that can be understood by nontechnical stakeholders. Class-based modelling represents: -

- The objects the system will manipulate
- The operation that will be applied for effective manipulation
- The relationships between the objects
- The collaborations that occur between the classes.

6.2 Identifying analysis classes

Classes are identified by underlining each noun or noun phrase and plotting it into a simple table. If the class (noun) is required to implement a solution, then it becomes a part of the solution space. Otherwise if the noun is used only to describe a solution, it is regarded as a part of the problem space. Once all the nouns have been isolated, General classification and Selection is done.

6.3 General classification

Nouns belonging to the solution space should exhibit any of the following criteria to be considered as a class. The 7 general characteristics are stated below:

1. **External entities** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. **Things** (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. **Occurrences or events** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. **Roles** (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. **Organizational units** (e.g., division, group, team) that are relevant to an application.
6. **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. **Structures** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

Serial	Noun	S/P	General Classification
1	Project	P	
2	Tution	P	
3	Guardian	**S**	4, 5
4	Student	**S**	4, 5
5	Tutor	**S**	4, 5
6	username	**S**	2
7	System	P	
8	Account	P	
9	FirstName	**S**	2

10	LastName	**S**	2
11	Address	**S**	2
12	Mobile Number	**S**	2
13	Email	**S**	2
14	Gender	**S**	2
15	Edu. Institute's name	**S**	2
16	Tutor's Subject	**S**	2
17	Current Position	**S**	2
18	Identity Card picture	**S**	2
19	Reference	**S**	2
20	Verification Code	P	
21	Password	**S**	2
22	Verified Tutor	**S**	4, 5
23	Authentication	**S**	4,5
24	Admin	**S**	4, 5
25	Authenticated Tutor	P	
26	Guardian Profile	S	2, 5, 7
27	Tutor Profile	**S**	2, 5, 7
28	Default Picture	P	
29	Profile Picture	**S**	2
30	Medium of Version	**S**	2
31	Preferred Classes	**S**	2
32	Preferred Group	**S**	2
33	Preferred Subject	**S**	2
34	Preferred Area	**S**	2
35	Experience Status	**S**	2

36	Minimum Salary	**S**	2
37	Notification	**S**	2, 3, 7
38	User	P	
39	Profile Info	P	
40	Utility Feature	P	
41	Calendar	**S**	1, 2
42	Reminder	**S**	1, 2
43	Notes	**S**	1, 2
44	Demo Video	S	2
45	Tuition Post	**S**	2, 5
46	Title of Post	**S**	2
47	Student Institute's name	**S**	2
48	Medium/Version	**S**	2
49	Student Class	**S**	2
50	Student Group	**S**	2
51	Subject List	**S**	2
52	Tutor Gender Preference	**S**	2
53	Days per Week	**S**	2
54	Student Area Address	**S**	2
55	Details Address	**S**	2
56	Contact Number	**S**	2
57	Salary Range	**S**	2
58	Others Option	**S**	2
59	Availability	**S**	2
60	Message Box	**S**	2, 3
61	Default Notification	P	

62	Reminder Notification	P	
63	Availability of Post	P	
64	Searching	S	3
65	Viewing	S	3
66	Initial Message	P	
67	Message Request	S	3
68	Group	**S**	5, 7
69	Group Name	**S**	2
70	Group Address	**S**	2
71	Group Image	**S**	2
72	Group Admin	**S**	4, 5
73	Batch	**S**	5, 7
74	Batch Name	**S**	2
75	Payment	**S**	2
76	Schedule	**S**	2
77	Seat's Availability	**S**	2
78	Academic Days	**S**	2
79	Batch Info	P	
80	StudentInfo	**S**	2
81	Student Name	**S**	2
82	Student Address	**S**	2
83	Student Phone Number	**S**	2
84	Student's Institute's Name	**S**	2
85	Student Class Position	**S**	2
86	Notice Board	S	2, 3
87	Notice	**S**	2

88	Post	**S**	2
89	Attachment	**S**	2
90	Files	**P**	
91	Blocking Option	**P**	

6.4 Selection criteria

The six selection characteristics that should be used as you consider each potential class for inclusion in the analysis model:

1. **Retained information**: The potential class will be useful during analysis only if information about it must be remembered so that the system can Function.

1. **Needed services:** The potential class must have a set of identifiable operations that can change the value of its attributes in some way.

2. **Multiple attributes**: During requirement analysis, the focus should be on “major” information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.

3. **Common attributes**: A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.

4. **Common operations**: A set of operations can be defined for the potential

class and these operations apply to all instances of the class.

5. **Essential requirements**: External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

Serial	Noun	Accepted Criteria
1	Guardian	1, 2, 3, 4, 5
2	Authentication	1, 2, 3, 4, 5
3	Student	1, 2, 3, 4, 5
4	Tutor	1, 2 , 3, 4, 5
5	Verified Tutor	1, 2, 3, 4, 5
6	Admin	1, 2, 3, 4, 5
7	Guardian Profile	1, 2, 3, 4, 5, 6
8	Tutor Profile	1, 2, 3, 4, 5, 6
9	Calendar	1, 2, 4, 6
10	Reminder	1, 2, 4, 6
11	Notes	1, 2, 4, 6
12	Tuition Post	1, 2, 3, 4, 5
13	Group	1, 2, 3, 4, 5
14	Group Admin	1, 2, 3, 4, 5
15	Batch	1, 2, 3, 4, 5
16	Notice Board	1, 2, 4, 5
17	Notification	2, 3, 6
18	Message Box	1, 2, 4, 5

6.5 Attribute and method selection

After identifying the classes, we have specified their attributes:

No	Name	Attributes	Method
1	Guardian	- username - address - mobileNumber	+signInWithMobileNumber() +verifiedMobileNumber() +addProfileInfo() +viewProfile() +editProfile() +searchTutorProfile() +searchGroup() +getSuggestion()
2	CandidateTutor	- firstName - lastName - address - mobileNumber - email - gender - institutionName - subject - currentPosition - identityCardPicture - reference	+signUpWithEmail() +verifyEmail() +signUpWithGoogleAccount() +signUpWithFacebookAccount() +addPersonalInfo() +addTheIdentityCardImage() +addReferences() +notifyTheReferenceEmail() +notifyAdmin()
3	VerifiedTutor	- profilePicture - preferredMedium - preferredClasses - preferredGroup - preferredSubjects - preferredAreas - experienceStatus	+addPreferenceInfo() +addProfilePicture() +viewProfile() +editProfile() +editPrivacyOfProfileInfo() +getSuggestion() +setAvailability()

		- minimumSalary - videoFile	+rateTutor()
4	Admin	- email - password -approvingId -blockingId	+approveTutor() +blockTutor()
5	Calendar	-eventTitle -eventLocation -eventDescription -eventDate -startingTime -endingTime -attendeesId	+viewAllEvents() +viewTheSelectedEvents() +createEvents() +editEvents() +deleteEvents() +joinInVideoChatWithGoogleMeet()
6	Notes	-noteName -noteText	+addNotes() +editNotes() +deleteNotes()
7	TuitionPost	-TitleOfPost - StudentInstituteName - Medium/Version - StudentClass - Group - SubjectList - TutorGenderPreference - DaysPerWeek - StudentAreaAddress - DetailAddress - ContactNumber - SalaryRange - OthersOption - Availability	+createNewPost() +editPost() +removePost() +setAvailability() +notifyPreferredArea'sTutor() +viewPost() +responseToPost() +notifyGuardianForTutorResponsing()
8	Group	- GroupName - GroupAddress - GroupImage	+createGroup() +editGroupInfo() +addTutor() +notifyGroupTutor()
9	Batch	- BatchName - Payment - Schedule - SeatAvailability - AcademicDays	+createBatch() +deleteBatch() +editBatchInfo() +addStudentInfo() +editStudentInfo() +deleteStudentInfo()

10	Notice Board	- NoticeTitle - Attachment	+addPost() +addAttachment() +deleteNotice()
11	MessageBox	-guardianMobileNumber - verifiedTutorEmail -messageRequestFromGuardian -messageRequestFromTutor -blockFromGuardianSide -blockFromTutorSide	+blockUser() +unblockUser() +sendMessageRequest() +acceptMessageRequest() +chat() +seenMessage() +readMessages() +selectImageForMessage() +uploadImage() +downloadImage()
12	DemoVideo	-videoId -videoName -videoUri	+uploadDemoVideo() +downloadVideo() +viewTheVideo() +deleteVideo()
13	Notification	-notificationId -notificationType -notificationTitle -notificationBody	+getNotifications() +viewNotification()

6.6 Finalizing classes

After analysis, we have found the following classes.

- Guardian
- CandidateTutor
- VerifiedTutor
- Admin
- Calendar
- Note
- TuitionPost
- MessageBox
- Group
- Batch
- NoticeBoard
- Notification
- DemoVideo

6.7 CRC Card

Guardian	
Attribute	**Method**
- username - address - mobileNumber	+signInWithMobileNumber() +verifiedMobileNumber() +addProfileInfo() +viewProfile() +editProfile() +searchTutorProfile() +searchGroup() +getSuggestion()
Responsibilities	**Collaborator**
1. Signing in 2. Profile making 3. Searching Tutors 4. Searching Groups 5. Messaging with Tutor 6. Reporting Tutors 7. Viewing Profile 8. Editing Profile 9. Viewing Tutors Profile 10. Getting notifications	VerifiedTutor TuitionPost MessageBox Group

CandidateTutor	
Attribute	**Method**
- firstName - lastName - address - mobileNumber - email - gender - institutionName - subject - currentPosition - identityCardPicture - reference	+signUpWithEmail() +verifyEmail() +signUpWithGoogleAccount() +signUpWithFacebookAccount() +addPersonalInfo() +addTheIdentityCardImage() +addReferences() +notifyTheReferenceEmail() +notifyAdmin()
Responsibilities	**Collaborator**
1. Signing Up 2. Authenticating Educational Institute	Authentication VerifiedTutor Admin

VerifiedTutor	
Attribute	**Method**
- profilePicture - preferredMedium/Version - preferredClasses - preferredGroup - preferredSubjects - preferredAreas	+addPreferenceInfo() +addProfilePicture() +viewProfile() +editProfile() +editThePrivacyOfProfileInfo() +getSuggestion() +setAvailability() +rateTutor() +reportTutor()

- experienceStatus - minimumSalary - videoFile	
Responsibilities	**Collaborator**
1. Adding Profile Info 2. Viewing Posts 3. Response to Posts 4. Viewing Guardian Profiles 5. Messaging with Guardians 6. Viewing own Profile 7. Editing Profiles 8. Viewing Notifications 9. Using utility features	Authentication Tutor Guardian Admin MessageBox Group Calendar Notes

Admin	
Attribute	**Method**
- email - password -approvingId -blockingId	+approveTutor() +blockTutor()
Responsibilities	**Collaborator**

1. Approving Tutors 2. Blocking Tutors	Authentication Tutor VerifiedTutor

Calendar	
Attribute	**Method**
-eventTitle -eventLocation -eventDescription -eventDate -startingTime -endingTime -attendeesId	+createEvents() +viewEvents() +editEvents() +deleteEvents() +joinInVideoChatWithGoogleMeet()
Responsibilities	**Collaborator**
1. Viewing Current Date 2. Create Events 3. Edit and Delete Events 4. Join In Video Chat With Google Meet.	VerifiedTutor

Notes	
Attribute	**Method**
-noteName -noteText	+addNotes() +editNotes() +deleteNotes()
Responsibilities	**Collaborator**
1. Adding notes 2. Viewing notes 3. Editing notes 4. Deleting notes	VerifiedTutor

TutionPost	
Attribute	**Method**
-TitleOfPost - StudentInstituteName - Medium/Version - StudentClass - Group - SubjectList - TutorGenderPreference - DaysPerWeek	+createNewPost() +editPost() +removePost() +setAvailability() +notifyPreferredArea'sTutor() +viewPost() +responseToPost() +notifyGuardianForTutorResponsing()

- StudentAreaAddress - DetailAddress - ContactNumber - SalaryRange - OthersOption - Availability	
Responsibilities	**Collaborator**
1. Creating a tuition post 2. Editing post 3. Removing post 4. Setting up availability of post	Guardian

MessageBox	
Attribute	**Method**
-guardianMobileNumber -verifiedTutorEmail -messageRequestFromGuardian -messageRequestFromTutor -blockFromGuardianSide -blockFromTutorSide	+blockUser() +unblockUser() +sendMessageRequest() +acceptMessageRequest() +chat() +seenMessage() +readMessages() +selectImageForMessage() +uploadImage() +downloadImage()
Responsibilities	**Collaborator**
1. Giving message request 2. Accepting message request	Guardian VerifiedTutor

3. Chatting 4. Blocking and Unblocking other users	Group

Group	
Attribute	**Method**
- GroupName - GroupAddress - GroupImage	+createGroup() +editGroupInfo() +addTutor() +notifyGroupTutor()
Responsibilities	**Collaborator**
1. Creating Group 2. Editing GroupInfo 3. Adding tutors to the Group 4. Viewing Group members 5. Viewing Group Info 6. Messaging with Group Admin	Guardian VerifiedTutor MassageBox Batch NoticeBoard

Batch	
Attribute	**Method**

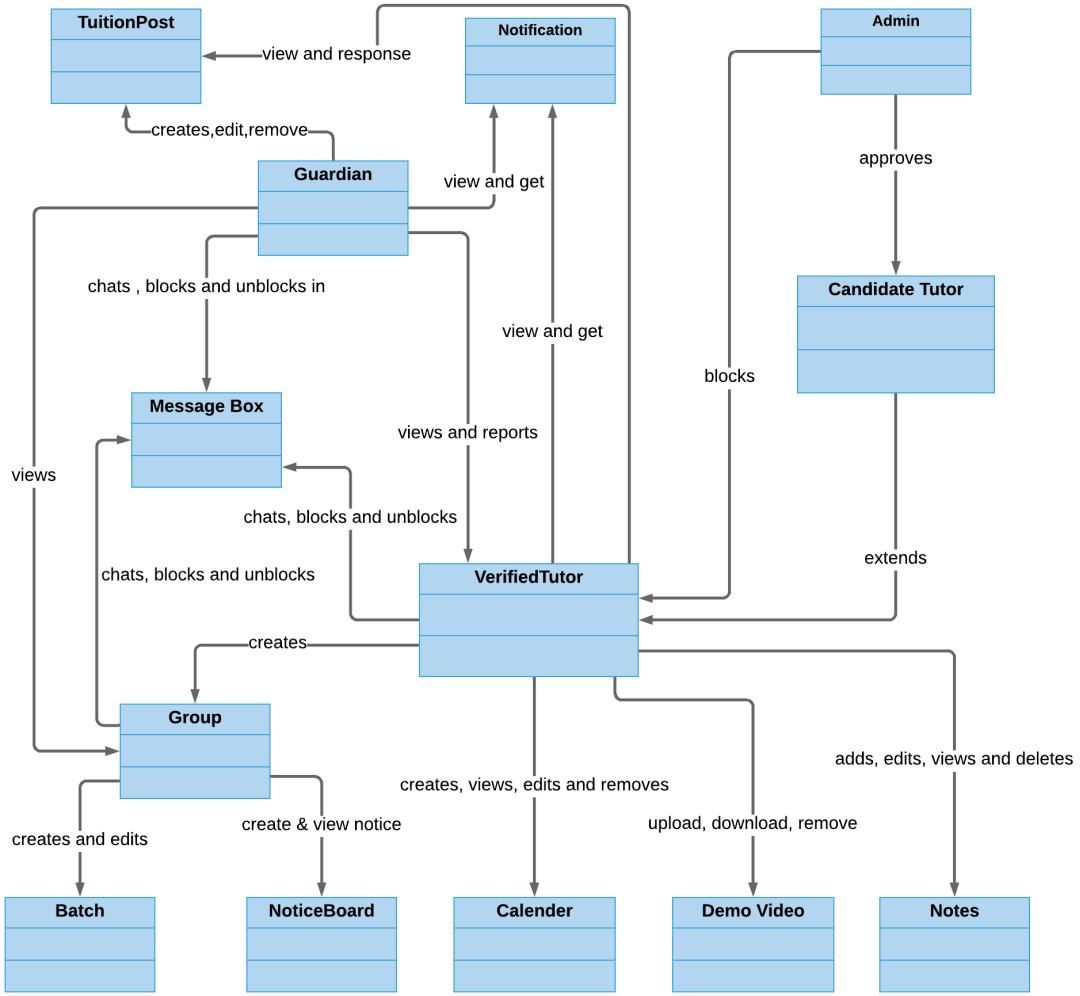
- BatchName - Payment - Schedule - SeatAvailability - AcademicDays	+createBatch() +deleteBatch() +editBatchInfo() +addStudentInfo() +editStudentInfo() +deleteStudentInfo()
Responsibilities	**Collaborator**
1. Creating Batch 2. Deleting Batch 3. Viewing Batch Info 4. Editing Batch Info 5. Adding Student Info 6. Editing Student Info 7. Deleting Student Info	Group

NoticeBoard	
Attribute	**Method**
- NoticeTitle - Attachment	+addPost() +addAttachment() +removeNotice()
Responsibilities	**Collaborator**

1. Creating notice 2. Adding Post 3. Adding Attachment	Group

Notification	
Attribute	**Method**
-notificationId -notificationType -notificationTitle -notificationBody	+getNotifications() +viewNotification()
Responsibilities	**Collaborator**
1. Get Notification 2. View Notification	Tutor Guardian

6.8 Class Diagram



CHAPTER 7

BEHAVIORAL MODELING

7.1 State Transition Diagram

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

7.1.1 Identifying Event:

Serial No	Events	Initiator	Collaborator	State Name
1	Provide tuition to the tutors	System	VerifiedTutor	ProvidingTuition
2	Provide tutors the guardian	System	Guardian	ProvidingTutor
3	Open an account	Tutor, Guardian	System	CreatingAccount
4	Code will be sent	System	Tutor,Guardian	SendingCode
5	Enter the verification code	Tutor, Guardian	System	EnteringCode
6	Verify the mobile number	System	Guardian	VerifyingNumber
7	Provide user info	Tutor, Guardian	System	AddingInfo
8	Checked student ID card photo	Admin	Tutor	AuthenticatingTutor

9	Provide id card photo	Tutor	System	ProvidingID
10	Referred to verified tutors	Tutor	System	ReferringTutor
11	Give email address of verified tutors as reference	Tutor	System	GivingReference
12	Approve the reference	VerifiedTutor	Tutor	ApprovingReference
13	Approve the tutor account	Admin	VerifiedTutor	ApprovingTutor
14	Report about fake info	Guardian	VerifiedTutor	ReportingId
15	Block Tutor	Admin	Guardian, VerifiedTutor	BlockingTutor
16	Add profile picture	Guardian, VerifiedTutor	System	AddingProfilePicture
17	Add Tutor profile info	VerifiedTutor	System	addingProfile
18	Set medium of version	VerifiedTutor	System	preferredVersion
19	Set preferred class	VerifiedTutor	System	PreferredClass
20	Set preferred subjects	VerifiedTutor	System	PreferredSubject
21	Set experienced status	VerifiedTutor	System	ExperiencedStatus
22	Set preferred areas	VerifiedTutor	System	PreferredAreas
23	Edit profile info	Guardian, Tutor	System	EditingProfile
24	Mark and unmark a date	Calendar	VerifiedTutor	MarkingDate
25	Set reminder	Reminder	VerifiedTutor	AddingReminder

26	Make notes	Notes	VerifiedTutor	AddingNotes
27	Upload demo videos	VerifiedTutor	System	UploadingVideo
28	Remove demo videos	VerifiedTutor	System	RemovingVideo
29	Create a post	Guardian	TuitionPost	CreatingPost
30	Communicate with guardian	VerifiedTutor	Guardian	GuardianCommunication
31	Communicate with tutor	Guardian	VerifiedTutor	TutorCommunication
32	Edit post	Guardian	TuitionPost	EditingPost
33	Remove post	Guardian	TuitionPost	RemovingPost
34	Set post availability	Guardian	TuitionPost	SettingPost
35	Get a notification	System	Guardian, VerifiedTutor	GettingNotification
36	Search the tutors	Guardian	System	SearchingTutor
37	Send an initial message	Guardian, VerifiedTutor	Guardian, VerifiedTutor	SendingMessage
38	Accept message request	Guardian, VerifiedTutor	Guardian, VerifiedTutor	AcceptingMessageRequest
39	Block in message box	Guardian, VerifiedTutor	Guardian, VerifiedTutor	BlockingMessage
40	Create group	VerifiedTutor	Group	CreatingGroup
41	Add tutors	VerifiedTutor	Group	AddingTutor
42	Edit group info	VerifiedTutor	Group	EditingGroup
43	Remove tutors	VerifiedTutor	Group	RemovingTutor
44	Create batch	VerifiedTutor	Group	CreatingBatch
45	Remove batch	VerifiedTutor	Group	RemovingBatch
46	Edit batch info	VerifiedTutor	Batch	EditingBatch

47	Add batch info	VerifiedTutor	Batch	AddingBatch
48	Remove batch info	VerifiedTutor	Batch	RemovingBatch
49	Store student info	VerifiedTutor	VerifiedTutor	StoringStudentInfo
50	Communicate with group	Guardian	VerifiedTutor	GroupCommunication
51	Create a notice	VerifiedTutor	NoticeBoard	CreatingNotice
52	Give post notification of preferred areas post	System	VerifiedTutor	postNotification
53	Give response notification	System	Guardian	responseNotification
54	Give availability notification	System	Guardian	availabilityNotification

7.1.2 Events after analysis:

After some analysis, we can merge some events and states which are of the same types. The analyzed data are given below:

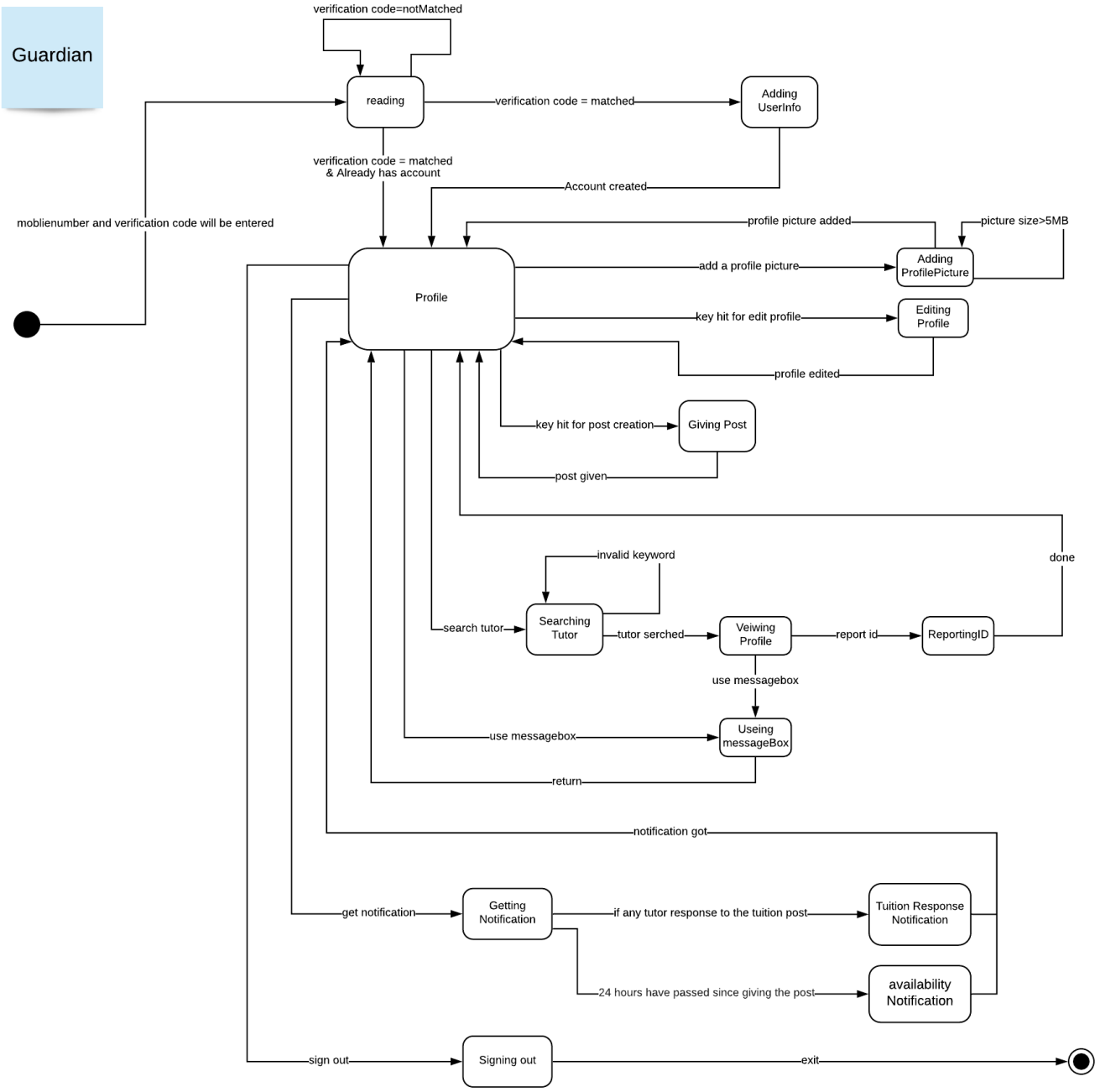
Serial No	**Events**	**Initiator**	**Collaborator**	**State Name**
1	Open an account	Tutor, Guardian	System	CreatingAccount
2	Sign in account	Tutor, VerifiedTutor, Guardian, Admin	System	SigningIn
3	Sign out of account	VerifiedTutor, Guardian,	System	SigningOut

		Admin		
4	Provide user info	Tutor, Guardian	System	AddingUserInfo
5	Checked student ID card photo	Admin	Tutor	CheckingIdCard
6	Provide id card photo	Tutor	System	ProvidingIDCardPhoto
7	Give email address of verified tutors as reference	Tutor	System	GivingReference
8	Approve the reference	VerifiedTutor	System	ApprovingReference
9	Approve the account	Admin	System	ApprovingTutor
10	Report about fake info	Guardian	VerifiedTutor	ReportingId
11	Block Tutor	Admin	VerifiedTutor	BlockingTutor
12	Add profile picture	Guardian, VerifiedTutor	System	AddingProfilePicture
13	Add Tutor profile info	VerifiedTutor	System	AddingProfile
14	Edit profile info	Guardian, VerifiedTutor	System	EditingProfile
15	Mark and Unmark a date	Calendar	VerifiedTutor	MarkingDate/UnmarkingDate
16	Set reminder	Reminder	VerifiedTutor	settingReminder
17	Make notes	Notes	VerifiedTutor	AddingNotes
18	Upload demo videos	VerifiedTutor	System	UploadingVideo
19	Remove demo videos	VerifiedTutor	System	RemovingVideo

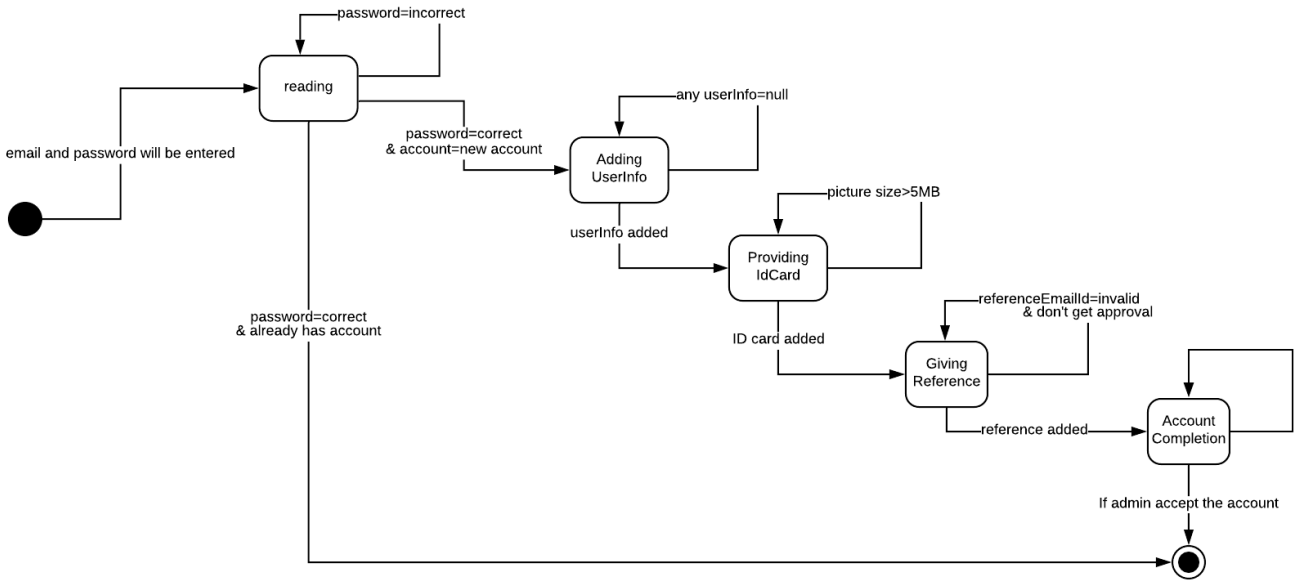
20	Create a post	Guardian	TuitionPost	CreatingPost
21	Communicate with guardian	VerifiedTutor	MessageBox	Chatting
22	Communicate with tutor	Guardian	MessageBox	Chatting
23	Edit post	Guardian	TuitionPost	EditingPost
24	Remove post	Guardian	TuitionPost	RemovingPost
25	Set post availability	Guardian	TuitionPost	SettingPostAvailability
26	Get a notification	System	Guardian, VerifiedTutor	GettingNotification
27	Search the tutors	Guardian	System	SearchingTutor
28	Send an initial message	Guardian, VerifiedTutor	MessageBox	SendingMessageRequest
29	Accept message request	Guardian, VerifiedTutor	MessageBox	AcceptingMessageRequest
30	Block in message box	Guardian, VerifiedTutor	MessageBox	BlockingMessage
31	Create group	VerifiedTutor	Group	CreatingGroup
32	Add tutors	Group	VerifiedTutor	AddingTutor
33	Edit group info	Group		EditingGroupInfo
34	Remove tutors	Group	VerifiedTutor	RemovingTutor
35	Create batch	Group	Batch	CreatingBatch
36	Remove batch	Group	Batch	RemovingBatch
37	Edit batch info	Batch		EditingBatchInfo
38	Add batch info	Batch		AddingBatchInfo
39	Remove batch info	Batch		RemovingBatchInfo
40	Store student info	Batch		StoringStudentInfo

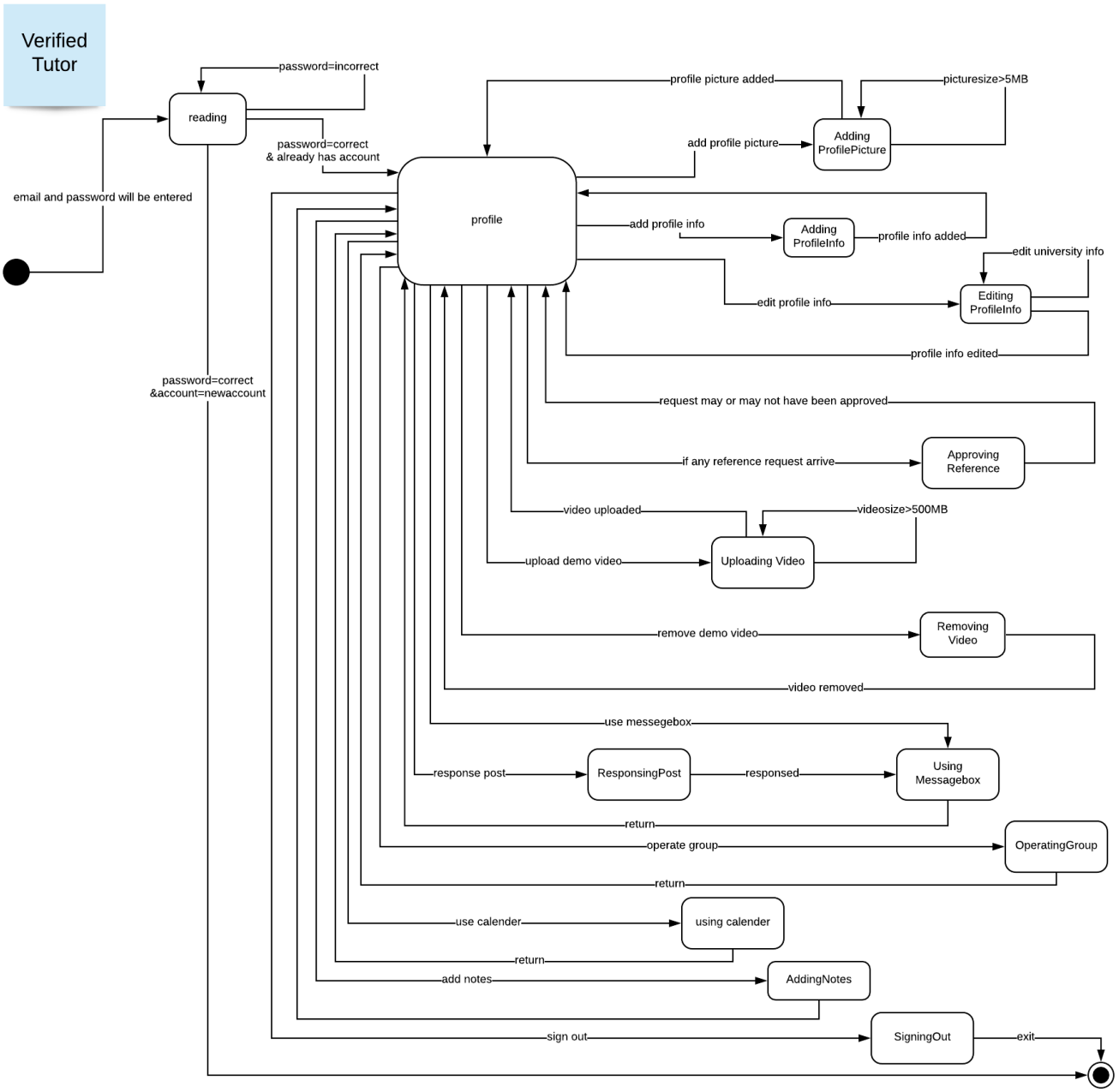
41	Communicate with group	Guardian	VerifiedTutor	GroupCommunication
42	Create a notice	Group	NoticeBoard	CreatingNotice
43	Give post notification of preferred areas post	System	VerifiedTutor	postNotification
44	Give response notification	System	Guardian	responseNotification
45	Give availability notification	System	Guardian	availabilityNotification

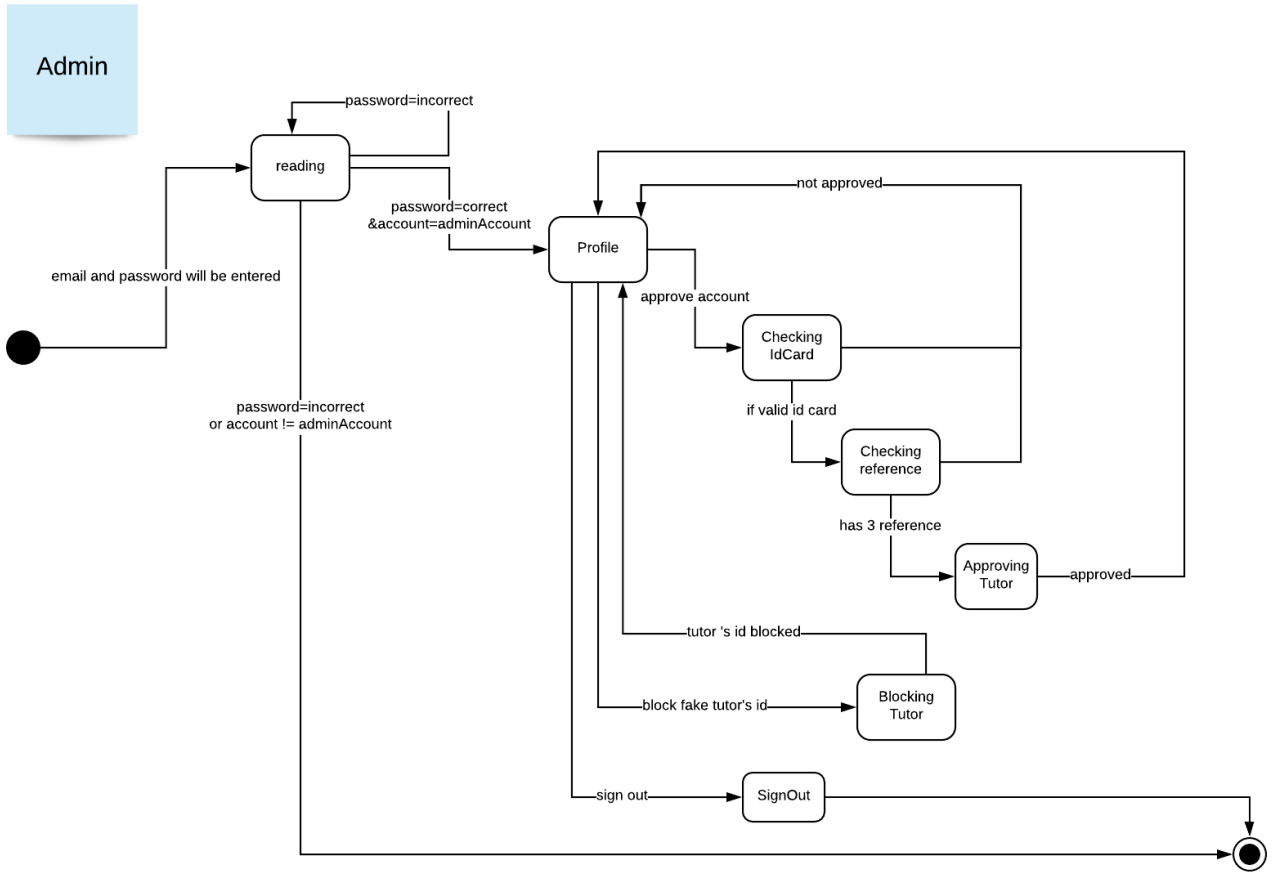
7.1.3 State Transition Diagram

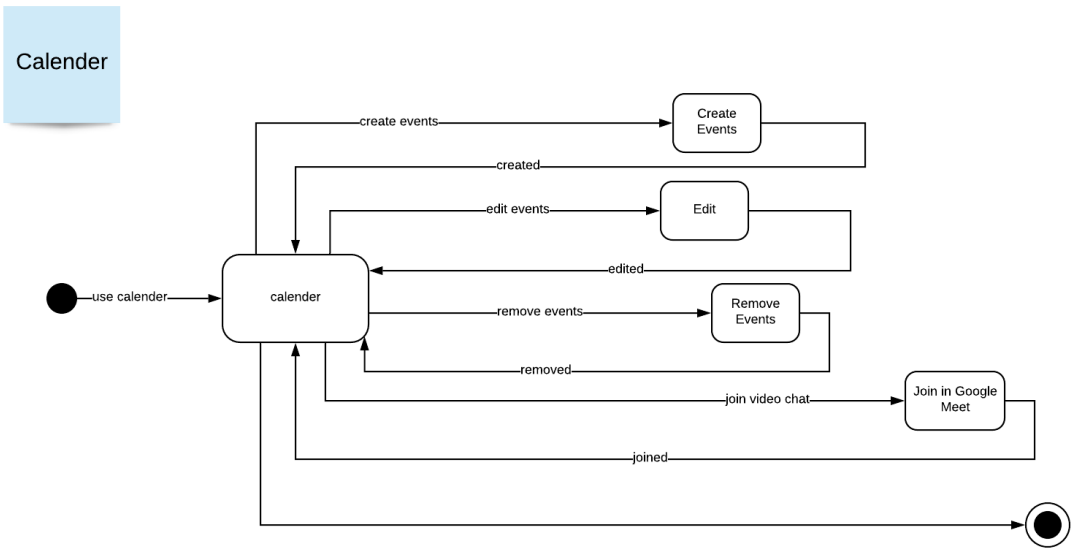

Guardian
verification code=notMatched
reading
verification code = matched
Adding UserInfo
verification code = matched & Already has account
Account created
moblienumber and verification code will be entered
profile picture added
picture size>5MB
Profile
add a profile picture
Adding ProfilePicture
key hit for edit profile
Editing Profile
profile edited
key hit for post creation
Giving Post
post given
done
invalid keyword
search tutor
Searching Tutor
tutor serched
Veiwing Profile
report id
ReportingID
use messagebox
Useing messageBox
use messagebox
return
notification got
get notification
Getting Notification
if any tutor response to the tuition post
Tuition Response Notification
24 hours have passed since giving the post
availability Notification
sign out
Signing out
exit

Tutor

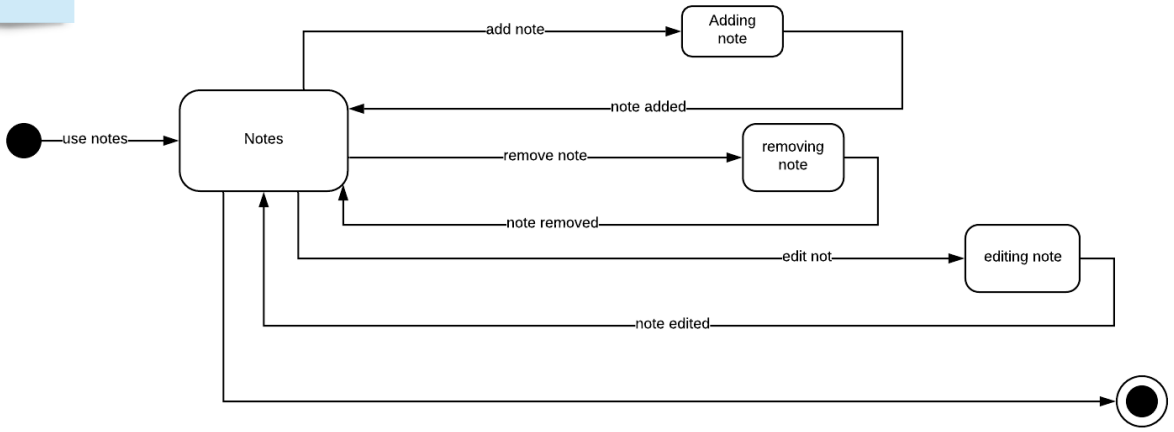

password=incorrect
reading
email and password will be entered
password=correct
& account=new account
any userInfo=null
Adding
UserInfo
userInfo added
picture size>5MB
Providing
IdCard
ID card added
referenceEmailId=invalid
& don't get approval
Giving
Reference
reference added
Account
Completion
If admin accept the account
password=correct
& already has account


Verified Tutor
password=incorrect
reading
password=correct
& already has account
email and password will be entered
password=correct
&account=newaccount
profile
profile picture added
picturesize>5MB
add profile picture
Adding ProfilePicture
add profile info
Adding ProfileInfo
profile info added
edit university info
edit profile info
Editing ProfileInfo
profile info edited
request may or may not have been approved
if any reference request arrive
Approving Reference
video uploaded
videosize>500MB
upload demo video
Uploading Video
remove demo video
Removing Video
video removed
use messegebox
response post
ResponsingPost
responsed
Using Messagebox
return
operate group
OperatingGroup
return
use calender
using calender
return
add notes
AddingNotes
sign out
SigningOut
exit

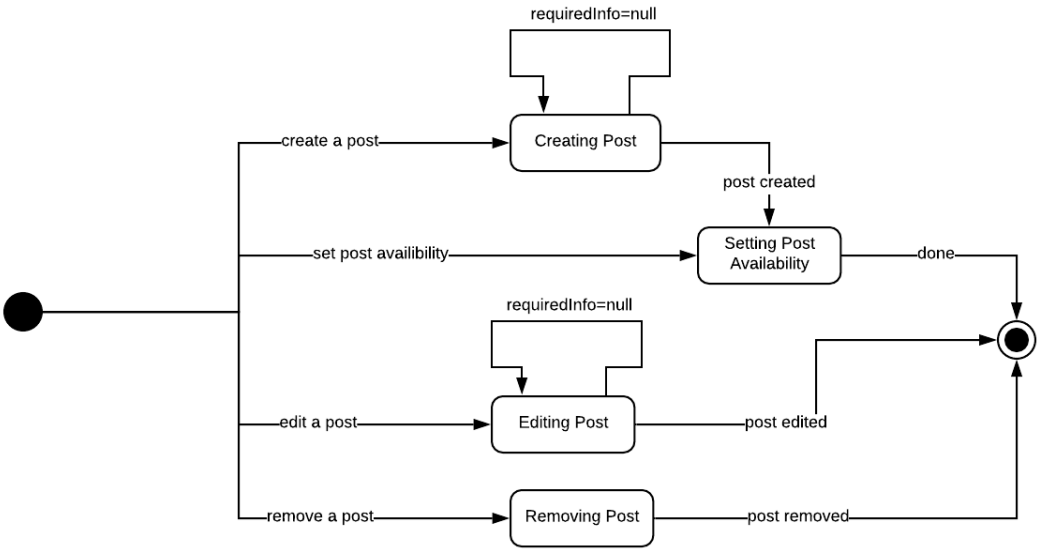

Admin
password=incorrect
reading
password=correct
&account=adminAccount
Profile
not approved
email and password will be entered
approve account
Checking
IdCard
if valid id card
password=incorrect
or account != adminAccount
Checking
reference
has 3 reference
Approving
Tutor
approved
tutor 's id blocked
Blocking
Tutor
block fake tutor's id
sign out
SignOut


Calender
create events
Create
Events
created
edit events
Edit
edited
use calender
calender
remove events
Remove
Events
removed
join video chat
Join in Google
Meet
joined

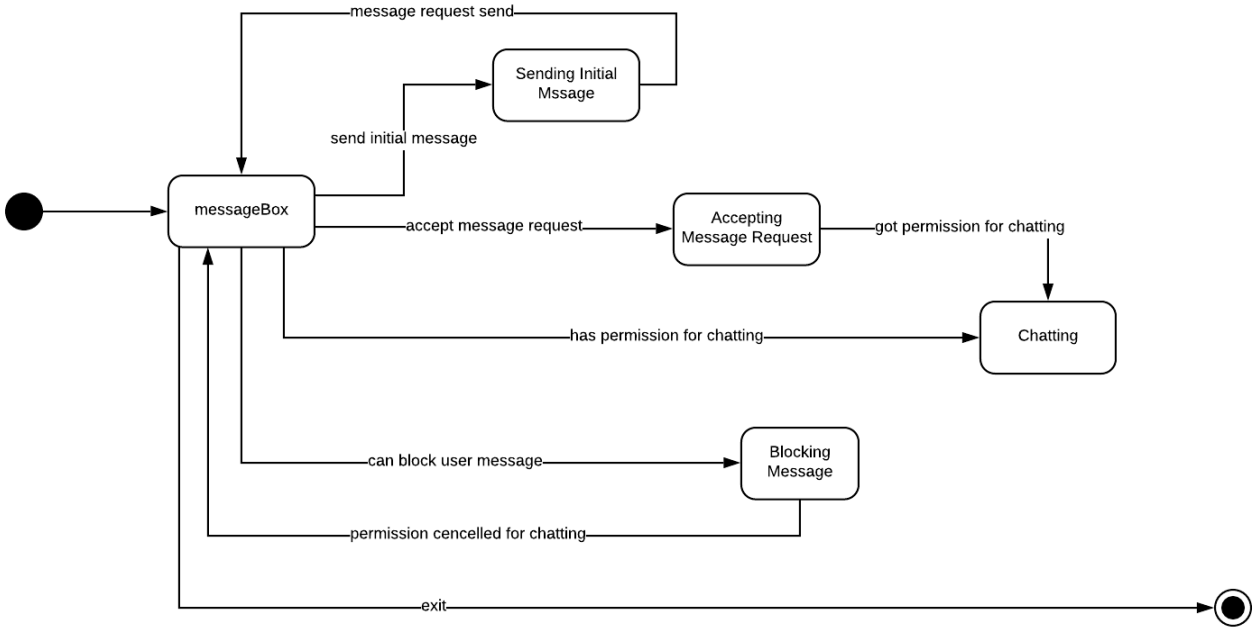
Notes



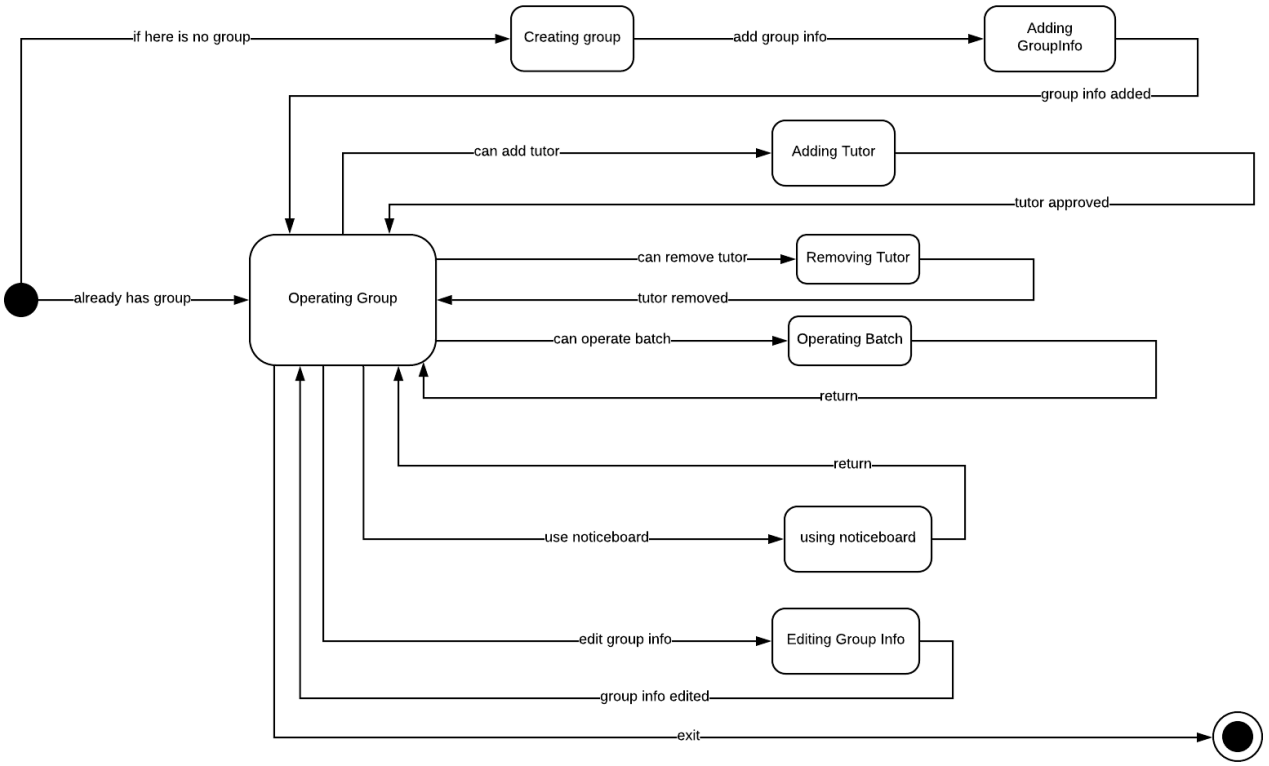
Tuition Post



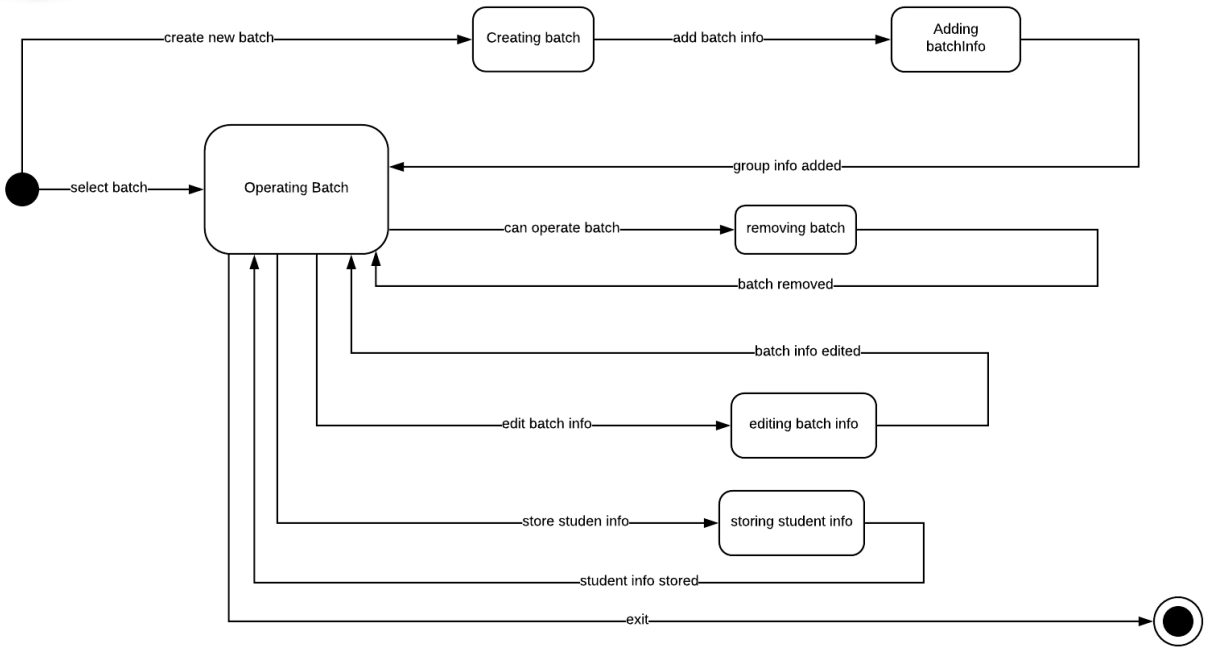
Message Box


message request send
Sending Initial Mssage
send initial message
messageBox
accept message request
Accepting Message Request
got permission for chatting
has permission for chatting
Chatting
can block user message
Blocking Message
permission cencelled for chatting
exit

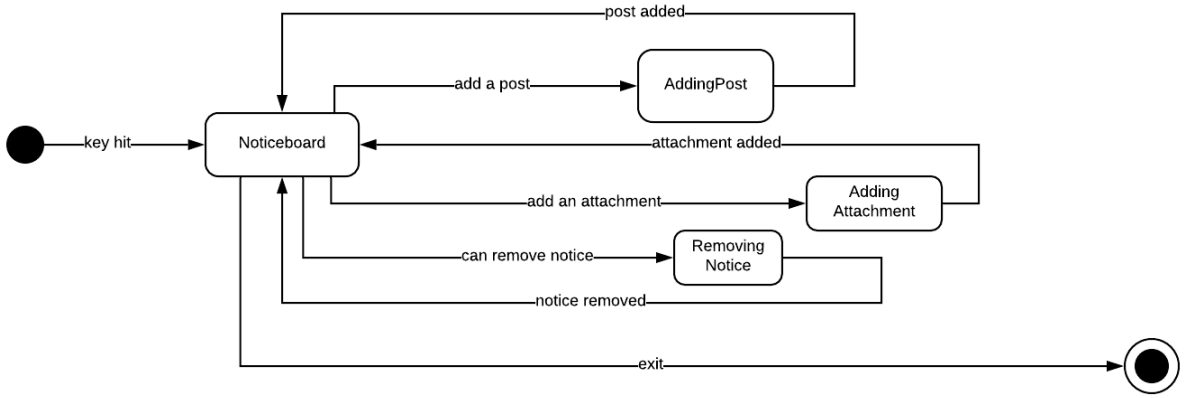
Group


if here is no group
Creating group
add group info
Adding GroupInfo
group info added
can add tutor
Adding Tutor
tutor approved
already has group
Operating Group
can remove tutor
Removing Tutor
tutor removed
can operate batch
Operating Batch
return
return
use noticeboard
using noticeboard
edit group info
Editing Group Info
group info edited
exit

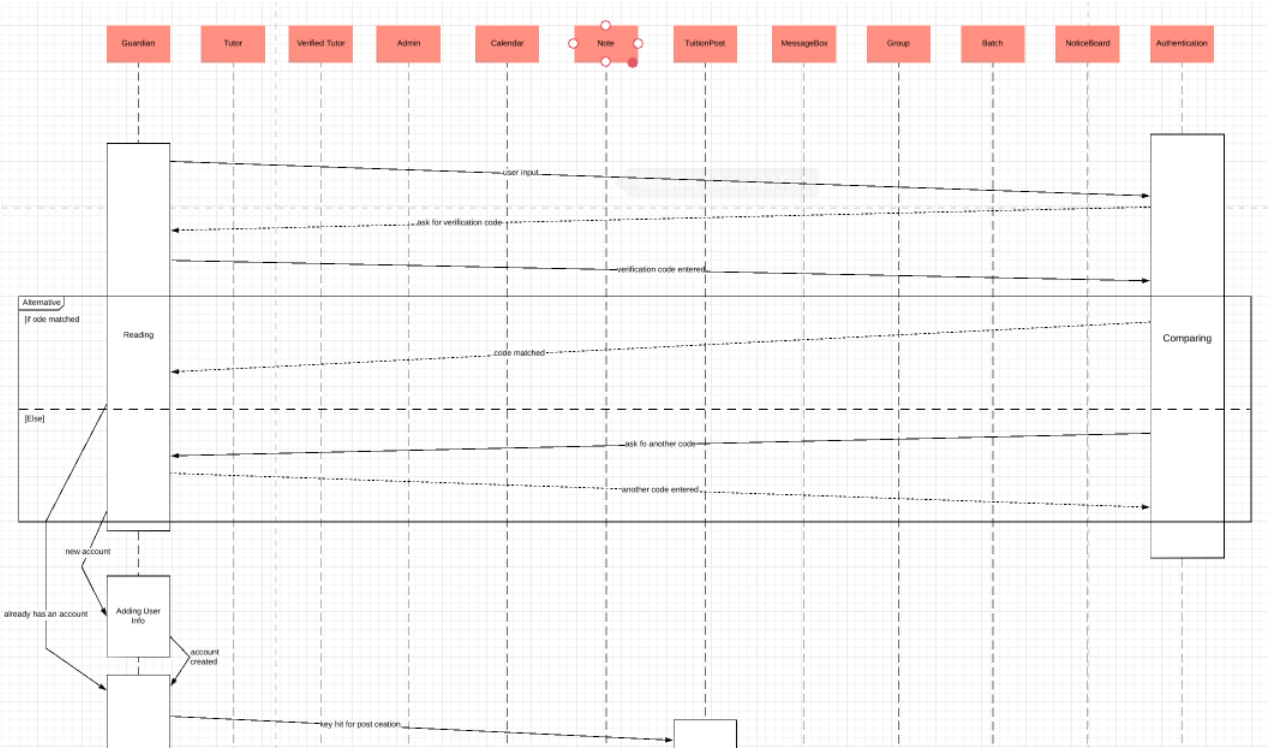
Batch

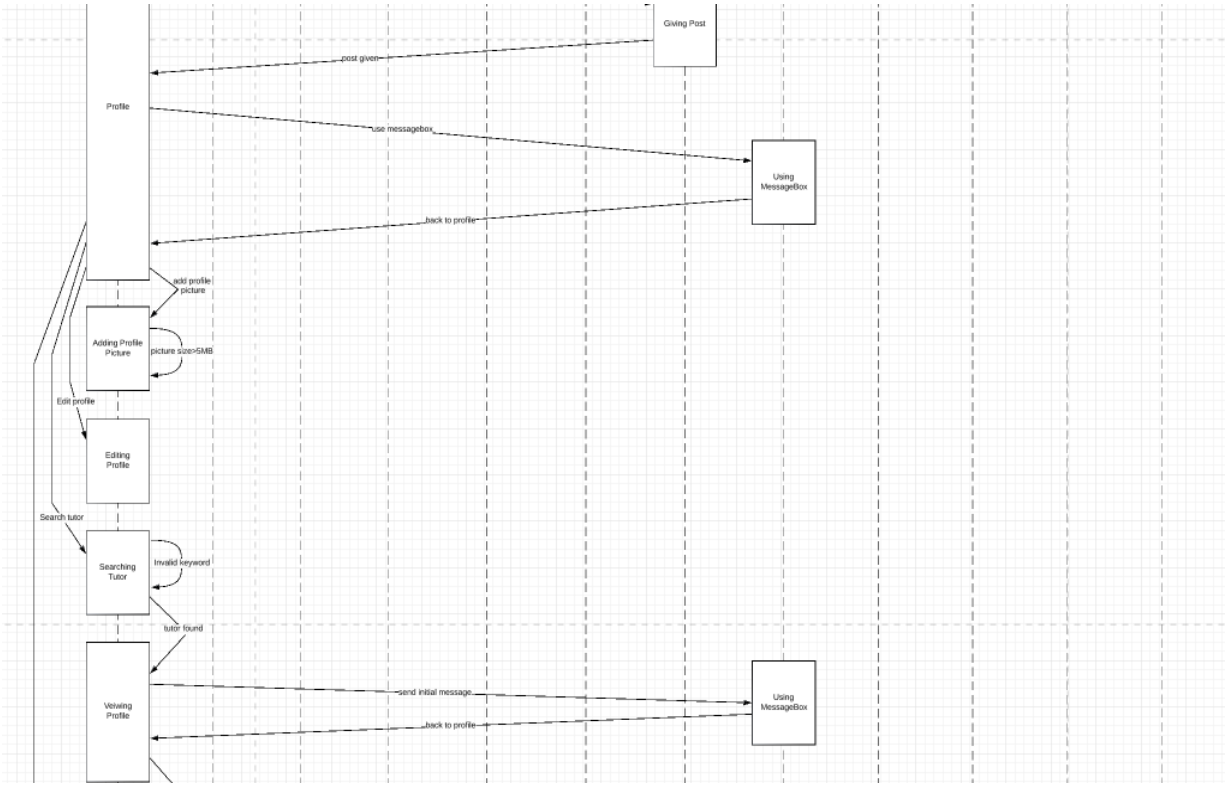


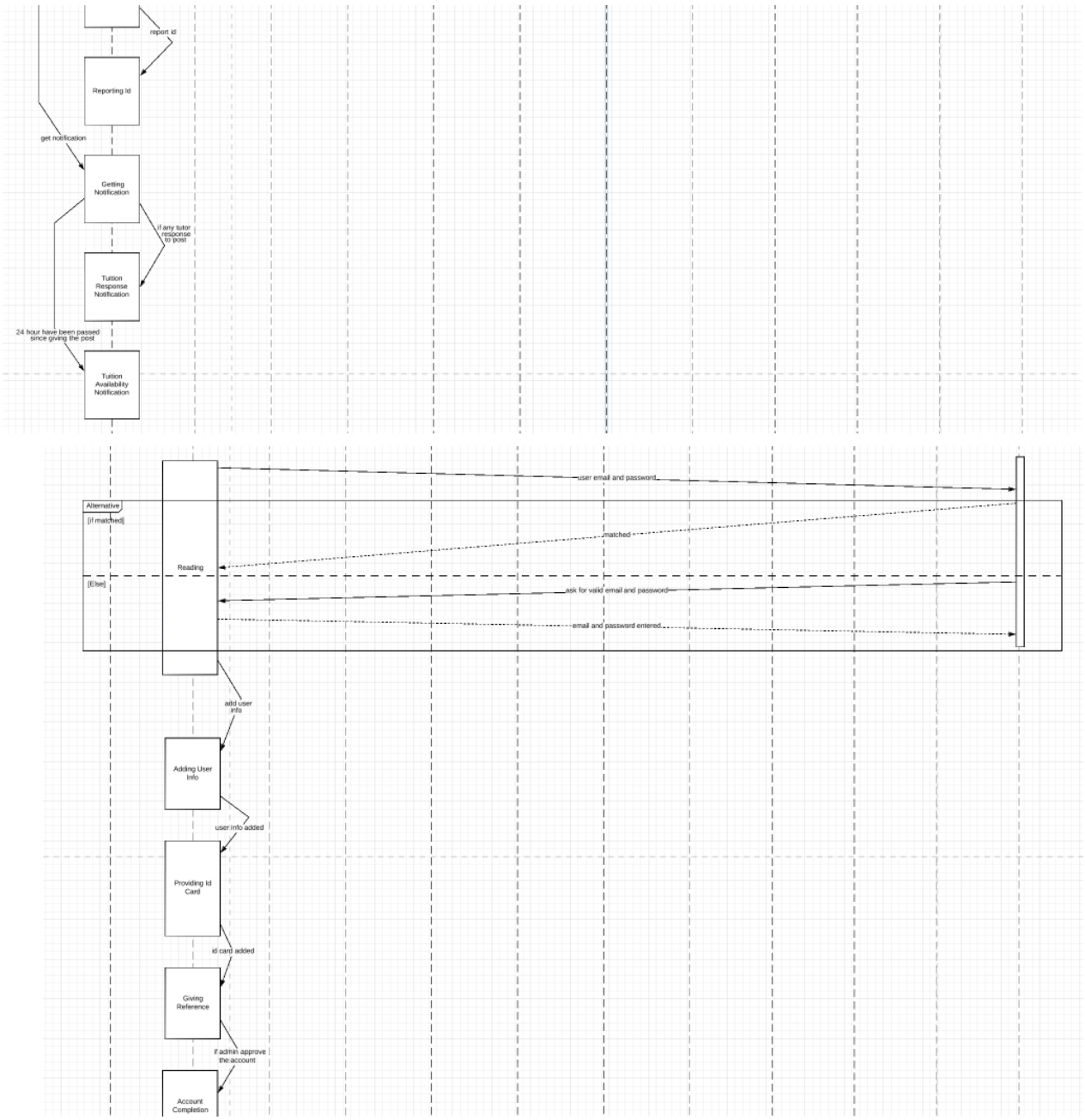
Notice

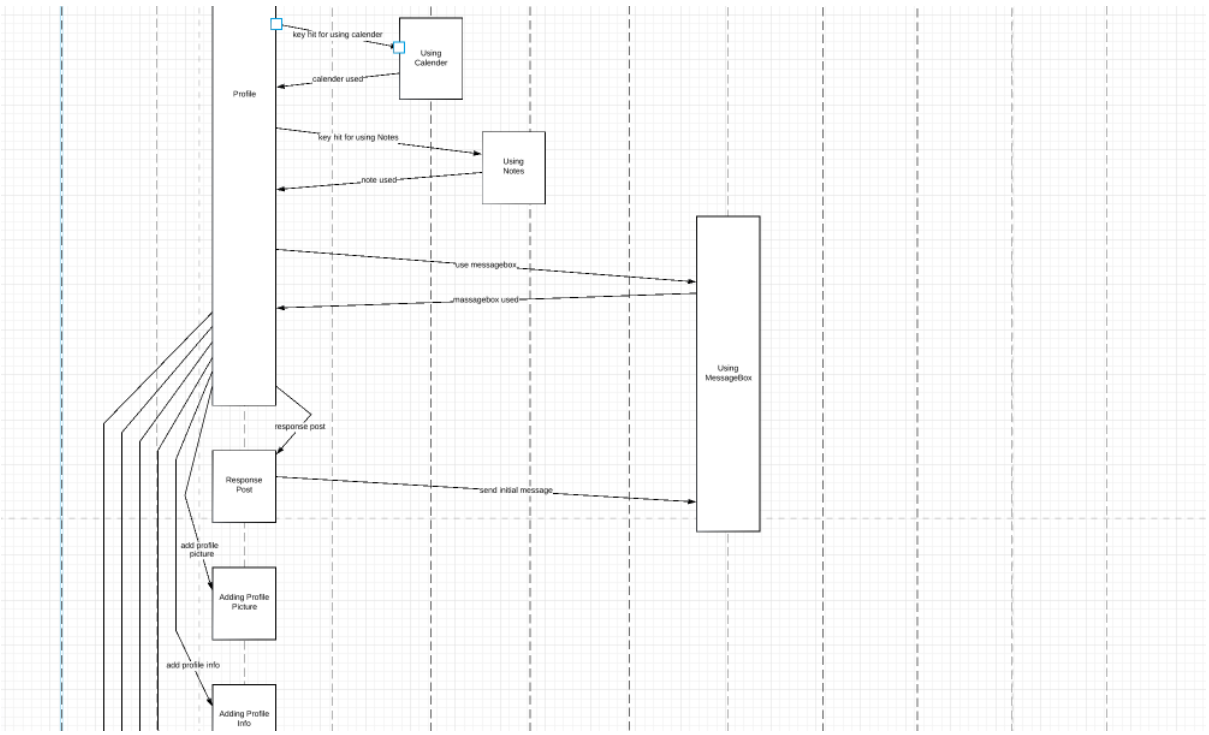


7.2 SEQUENCE DIAGRAM

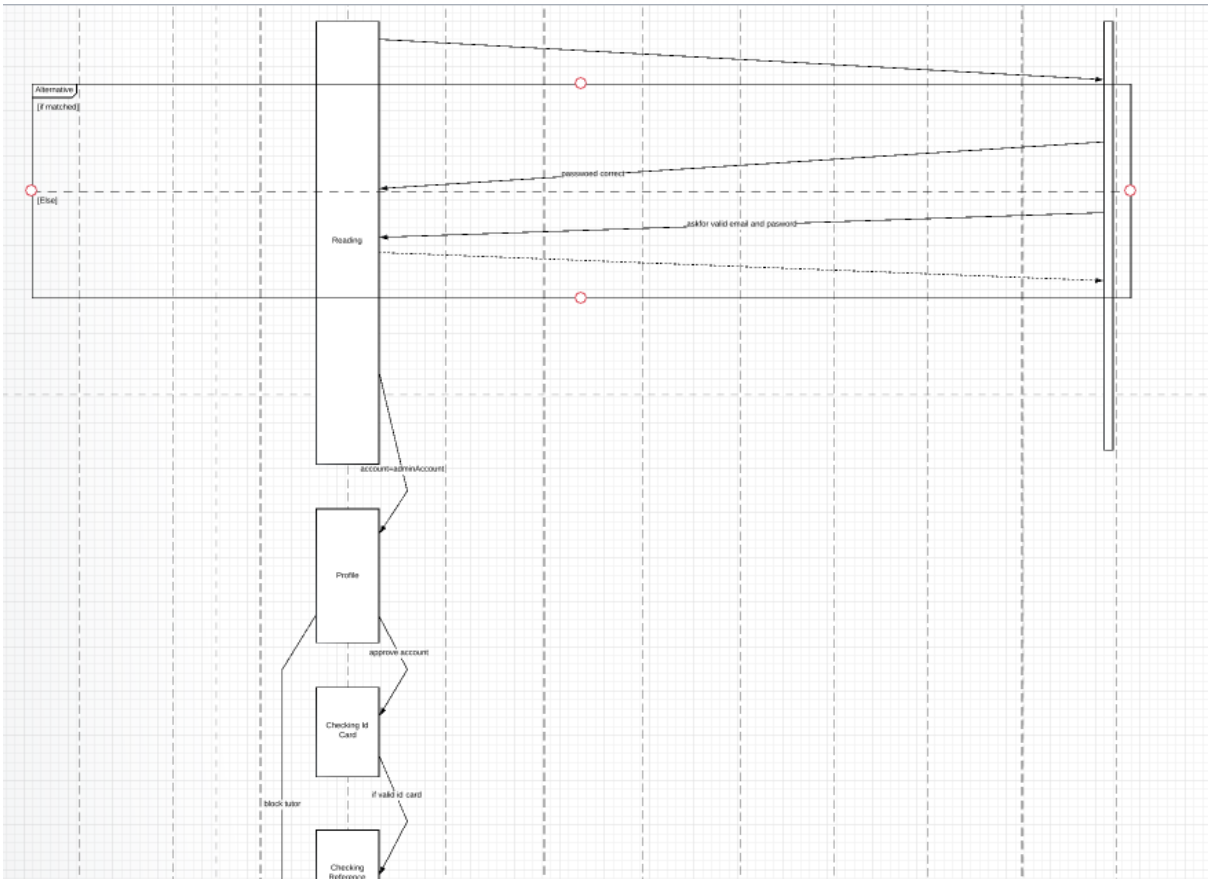


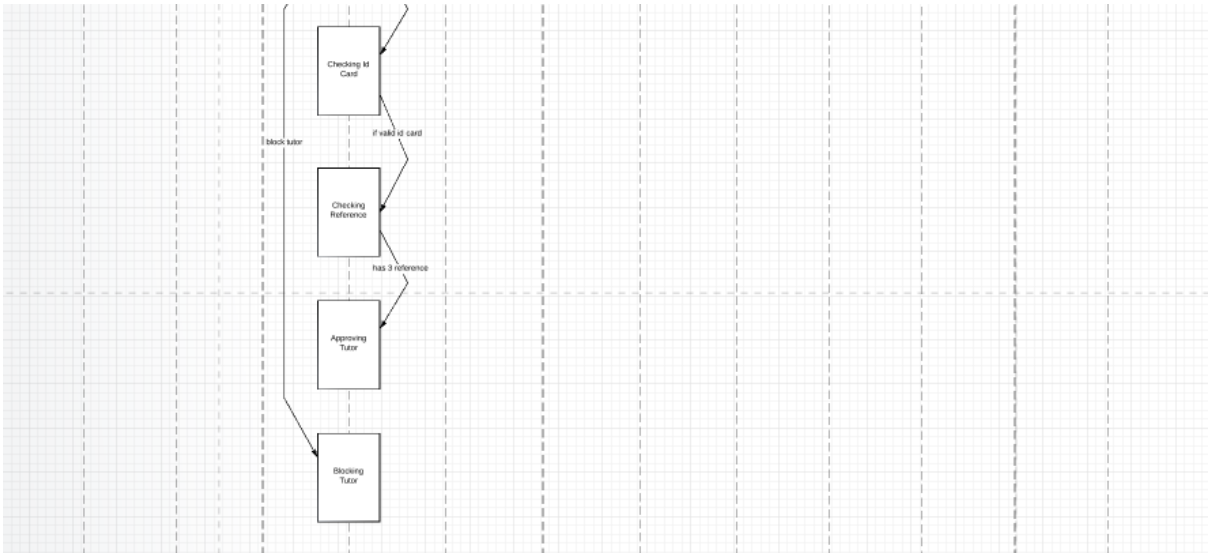


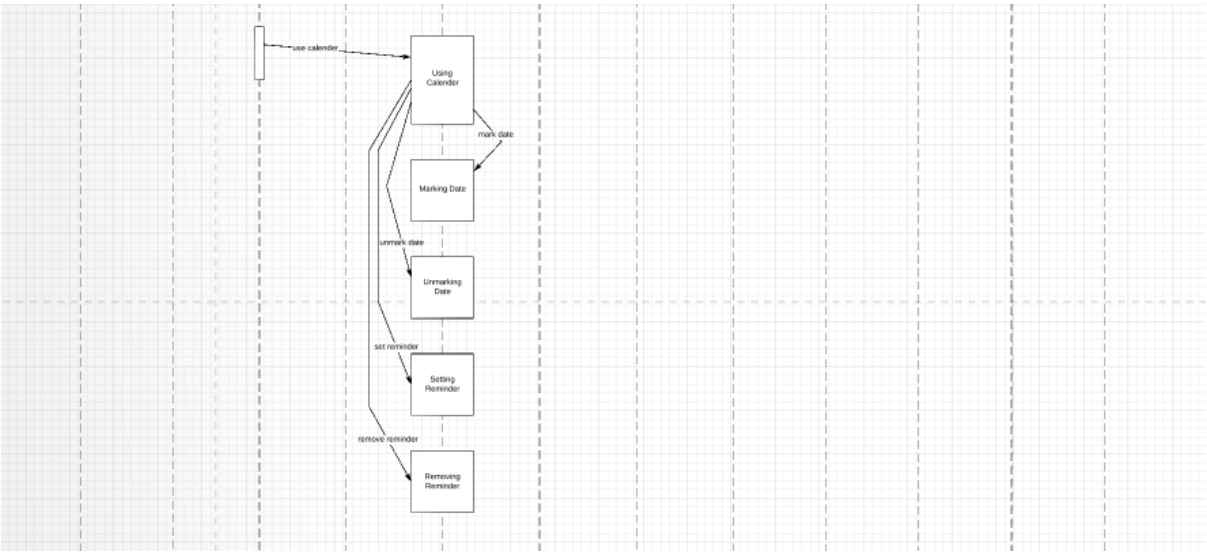

report id
Reporting Id
get notification
Getting Notification
if any tutor response to post
Tuition Response Notification
24 hour have been passed since giving the post
Tuition Availability Notification
user email and password
Alternative
[if matched]
matched
Reading
[Else]
ask for valid email and password
email and password entered
add user info
Adding User Info
user info added
Providing Id Card
id card added
Giving Reference
if admin approve the account
Account Completion

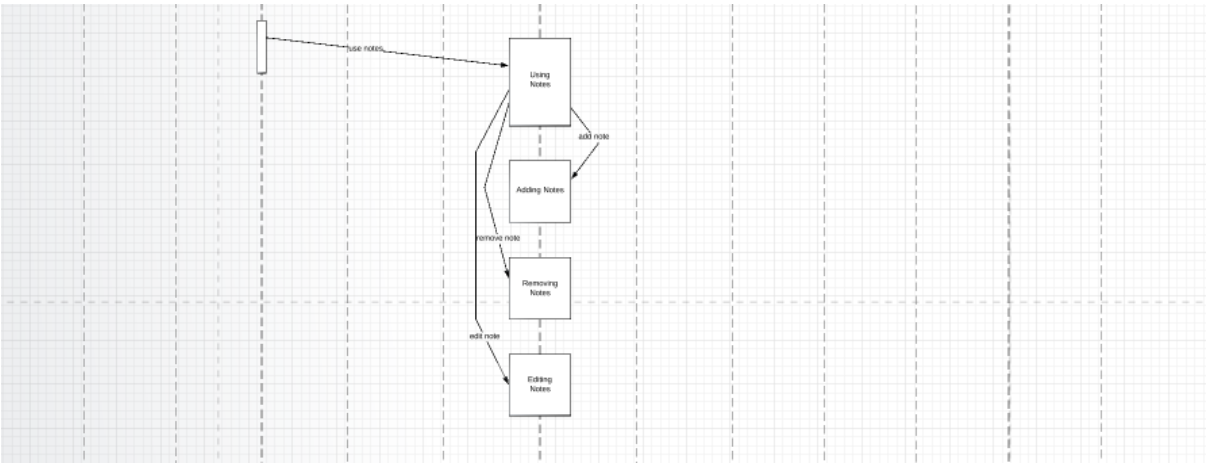

key hit for using calender
Using Calender
calender used
Profile
key hit for using Notes
Using Notes
note used
use messagebox
massagebox used
Using MessageBox
response post
Response Post
send initial message
add profile picture
Adding Profile Picture
add profile info
Adding Profile Info

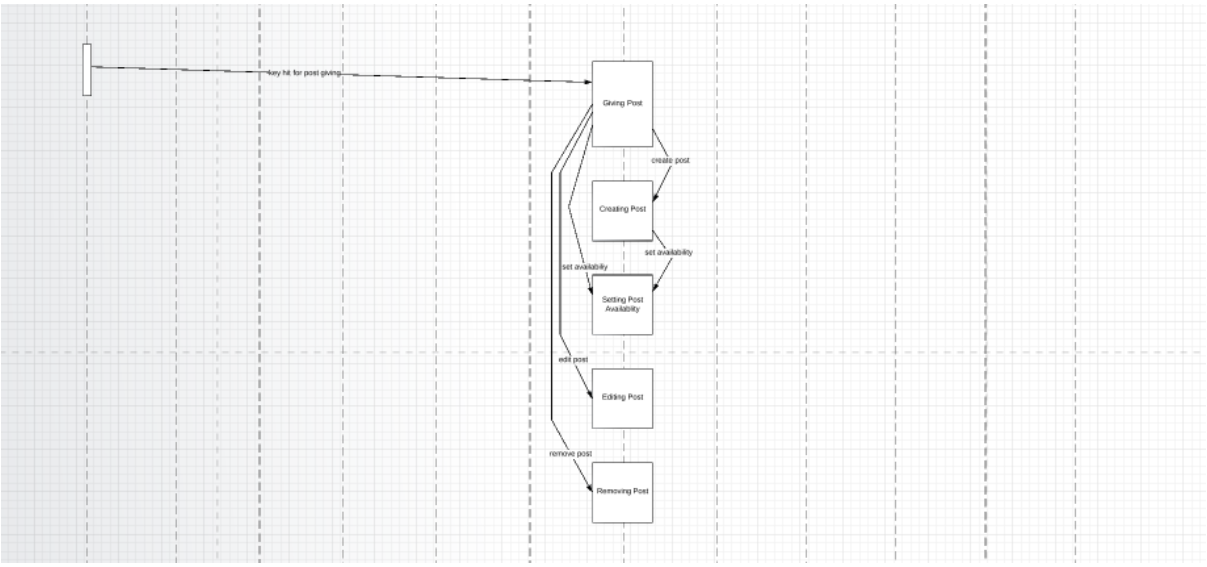

Adding Profile Info
edit profile info
Editing Profile Info
upload demo video
Uploading Demo Video
remove demo video
Removing Demo Video
approve reference
Approving Reference

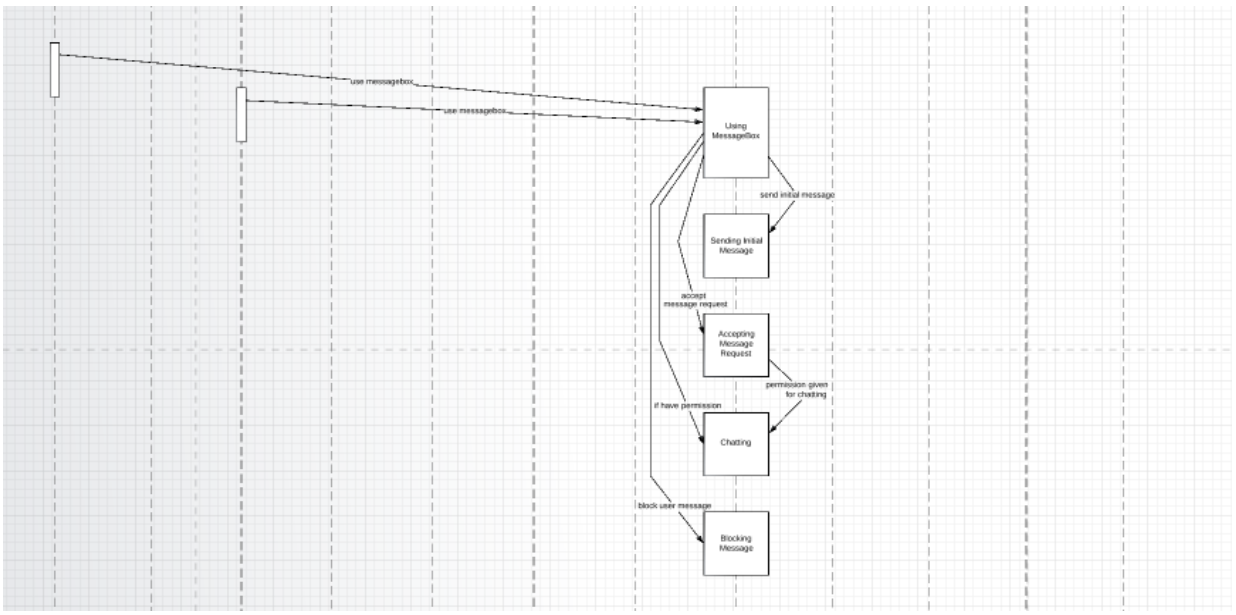

Alternative
[if matched]
[Else]
password correct
ask for valid email and paseword
Reading
account=adminAccount
Profile
approve account
Checking Id Card
if valid id card
block tutor
Checking Reference

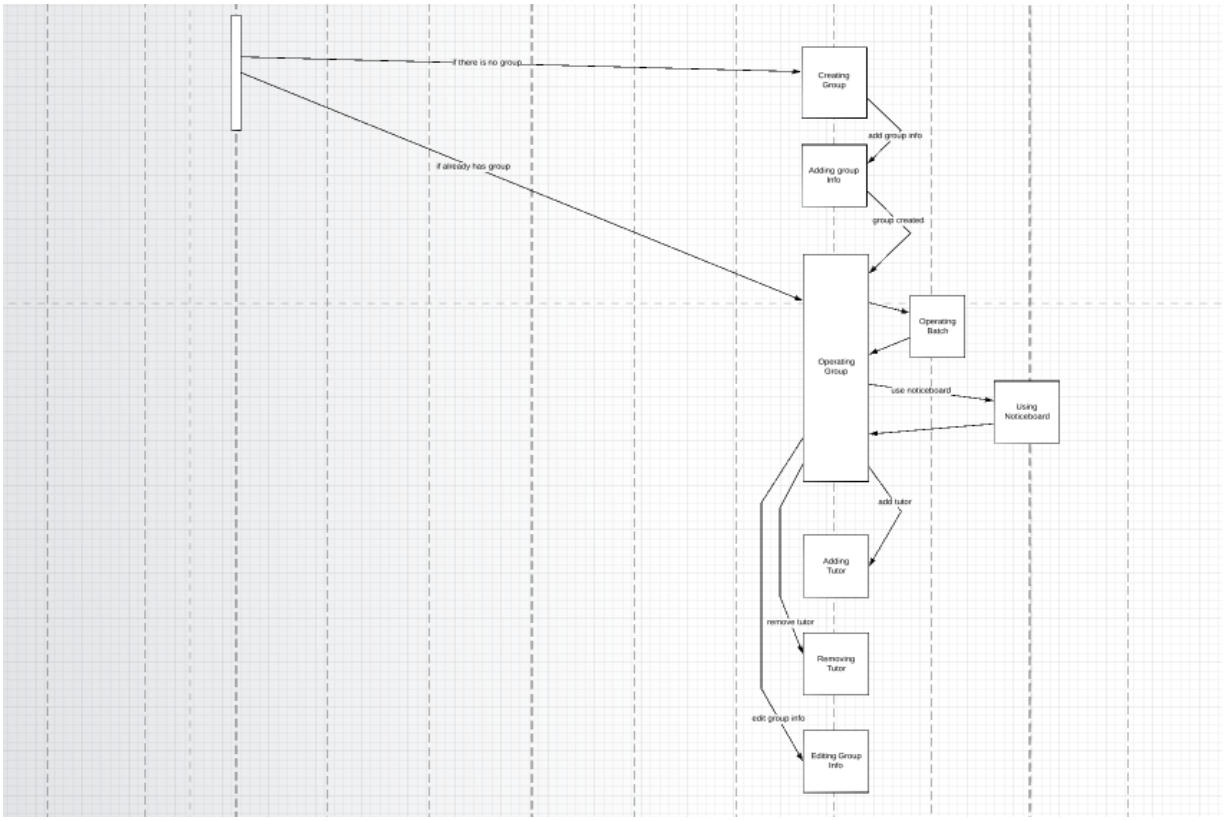

Checking Id Card
if valid id card
block tutor
Checking Reference
has 3 reference
Approving Tutor
Blocking Tutor

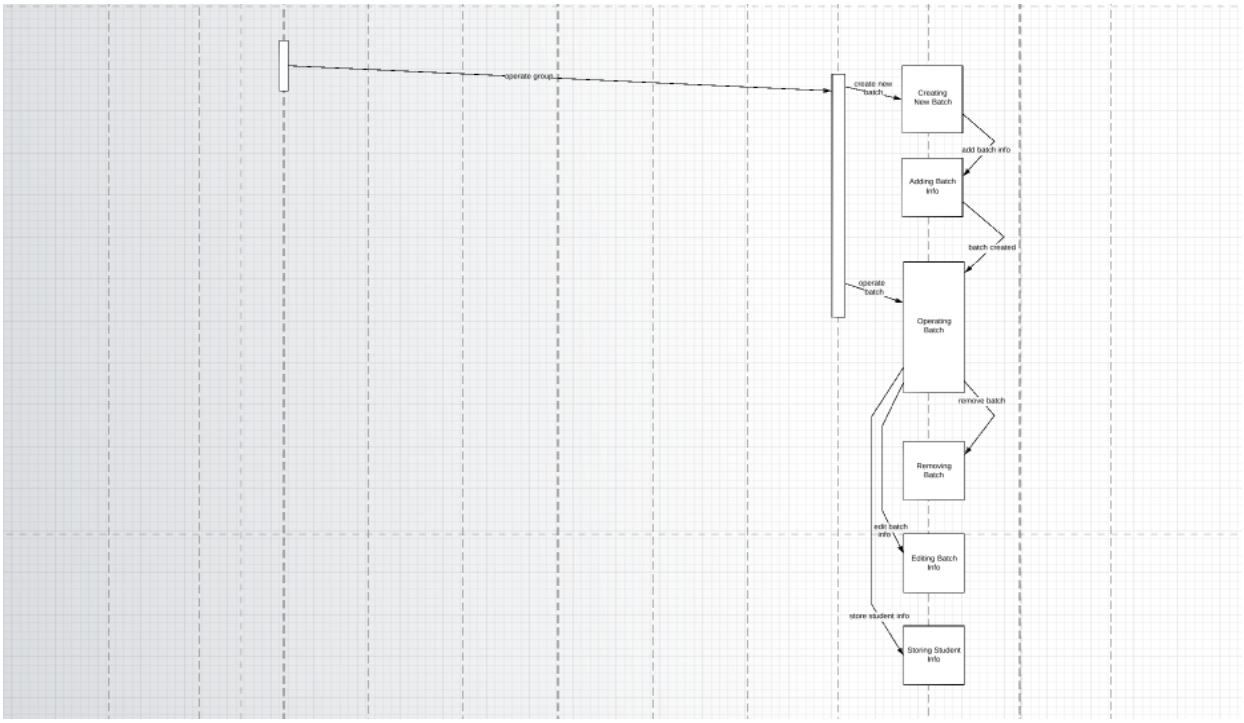

use calender
Using Calender
mark date
Marking Date
unmark date
Unmarking Date
set reminder
Setting Reminder
remove reminder
Removing Reminder

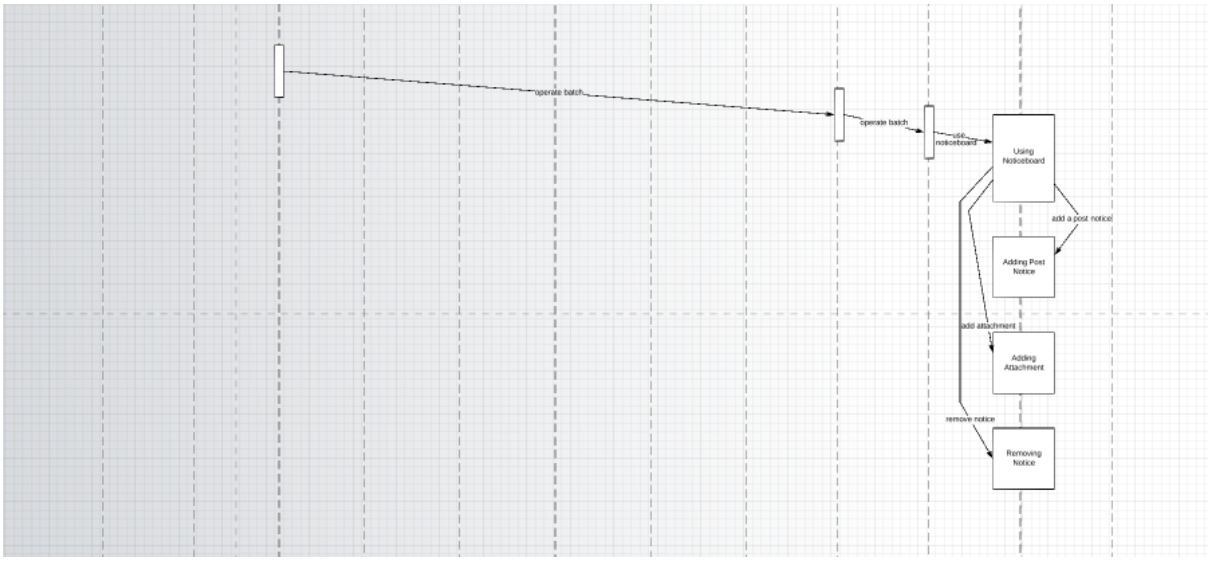

use notes
Using Notes
add note
Adding Notes
remove note
Removing Notes
edit note
Editing Notes


key hit for post giving
Giving Post
create post
Creating Post
set availability
set availability
Setting Post Availability
edit post
Editing Post
remove post
Removing Post


use messagebox
use messagebox
Using MessageBox
send initial message
Sending Initial Message
accept message request
Accepting Message Request
permission given for chatting
if have permission
Chatting
block user message
Blocking Message


if there is no group
if already has group
Creating Group
add group info
Adding group Info
group created
Operating Batch
Operating Group
use noticeboard
Using Noticeboard
add tutor
Adding Tutor
remove tutor
Removing Tutor
edit group info
Editing Group Info


operate group
create new batch
Creating New Batch
add batch info
Adding Batch Info
batch created
operate batch
Operating Batch
remove batch
Removing Batch
edit batch info
Editing Batch Info
store student info
Storing Student Info


operate batch
operate batch
use noticeboard
Using Noticeboard
add a post notice
Adding Post Notice
add attachment
Adding Attachment
remove notice
Removing Notice

CHAPTER 8

DATA FLOW MODELING

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

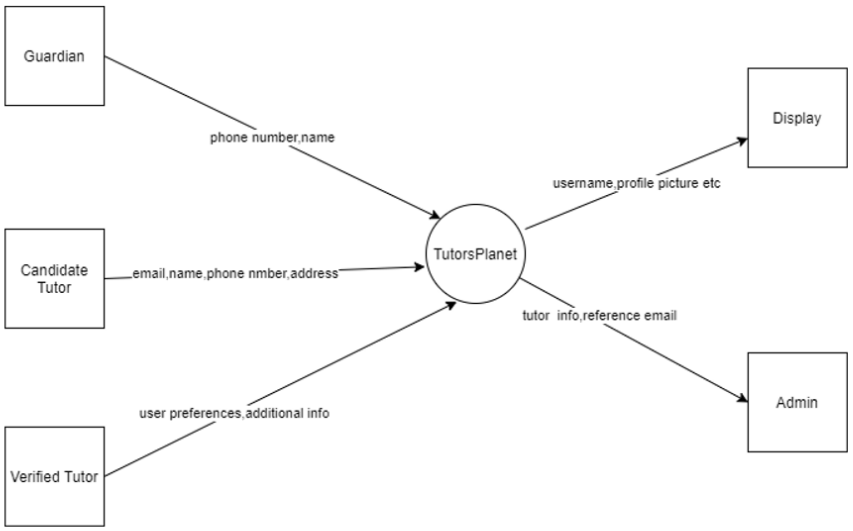


Figure: Level 0 Data Flow Diagram (Context Diagram)

Level 1

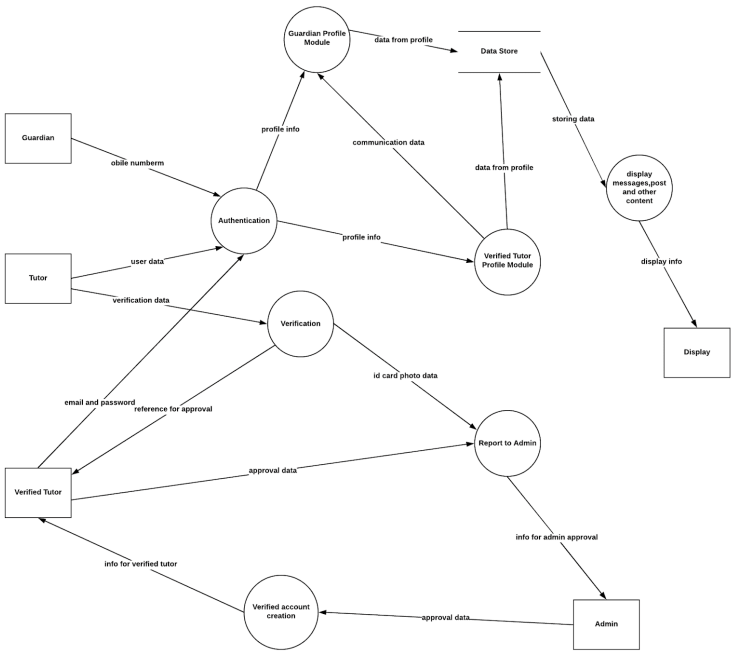


Figure: Level 1 Data Flow Diagram

Level 2
"Guardian Profile Module"

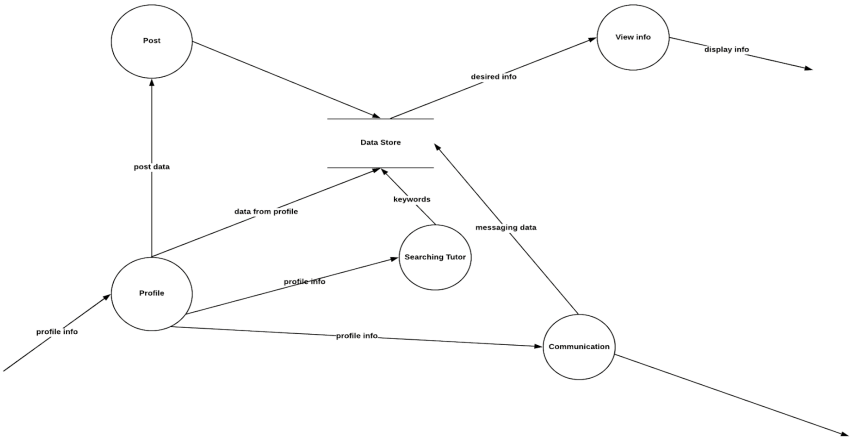


Figure: Level 2 Data Flow Diagram “Guardian Profile Module”

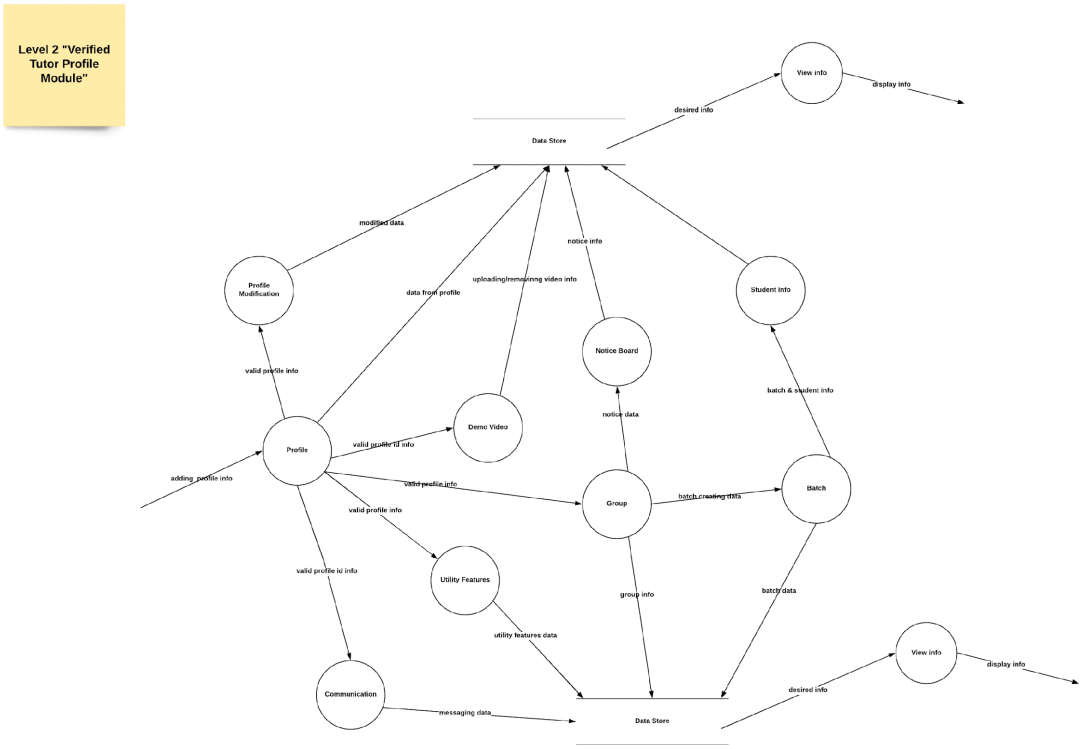


Figure: Level 2 Data Flow Diagram “Verified Tutor Profile Module”

CHAPTER 9

CONCLUSION

We are pleased to submit the final SRS report on this project “Tutors Planet”. From this, the readers will get a clear and easy view of the overall system of “Tutors Planet” application. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using this SRS. Hopefully, this document can also help our junior BSSE batch students. We tried our best to remove all dependencies and make an effective and fully designed SRS. We believe that the reader will find it in order.