

A Project Report On

“Online Tutor Finder System”

Submitted by: **Jagruti Ravindra Ahire**

Guided By: **Samadhan Khairnar**

Table of Contents

Chapter 1: Abstract

Chapter 2: Requirement Specification

Chapter 3: System Analysis and Design

3.1 ER Diagram

3.2 Use Case Diagram

3.3 Activity Diagram

3.4 State Diagram

3.5 Data Flow Diagram

Chapter 4: Data Dictionary

Chapter 1: Abstract

The system of private tuition has been in existence in India for a long time but in recent times it has grown in manifold affecting the core of the educational system. This Online Tutor Finder system will help to find tuition teachers from nearby locations. Teachers can also get a student just by logging onto the website and setting up the profile. This system can be used anywhere any time. In the Personal Tutor Finder system, there are three entities namely, Admin, Students, and Tutor. Admin can login, manage tutor by adding new teachers and update their profiles. Admin can also manage E-books by adding new books to the library. Admin can also check for the registered students. Students can register and login, tutors can be viewed by students. Student can filter and select the tutor and after selecting students will raise the request of the demo lecture. After attending the lecture, they can book the tutor online, rate the tutor and view the E-books. The tutor can check for the request for a demo lecture and accept the request. They can also check the booking done. They need to set their profile. The Personal Tutor Finder system can help the tutors to get students to find the best tutors.

Chapter 2: Requirement Specification

Functional Requirements: The system comprises of 3 major modules with their sub-modules as follows:

1. Admin:

- Login: Admin can login using credentials.
- Add Tutor: Admin can add new tutor and tutor's information.
- Manage Tutor: Admin can upload, delete, approve the records of tutors.
- View Tutor: Admin can check all updated information about tutor.
- Manage E Books: Admin can add E books.
- View Students: Admin can check register students.
- Manage Students: Admin can take or reject registration requests of students.
- View Rate: Admin can check all rated given by students.

2. Tutor:

- Registration: Tutor can sign up.
- Login: Tutor can login using credentials.
- Manage Reservation Requests: Tutors can accept or reject the requests of students.
- View or Access E-Books: Tutor can get access and study books from the library.
- View Ratings: Tutor can view student's scores.
- Upload Demo Lecture: Tutor can update, delete, create demo lecture schedule.
- Manage Students: Tutor can manage student's information.

3. Student:

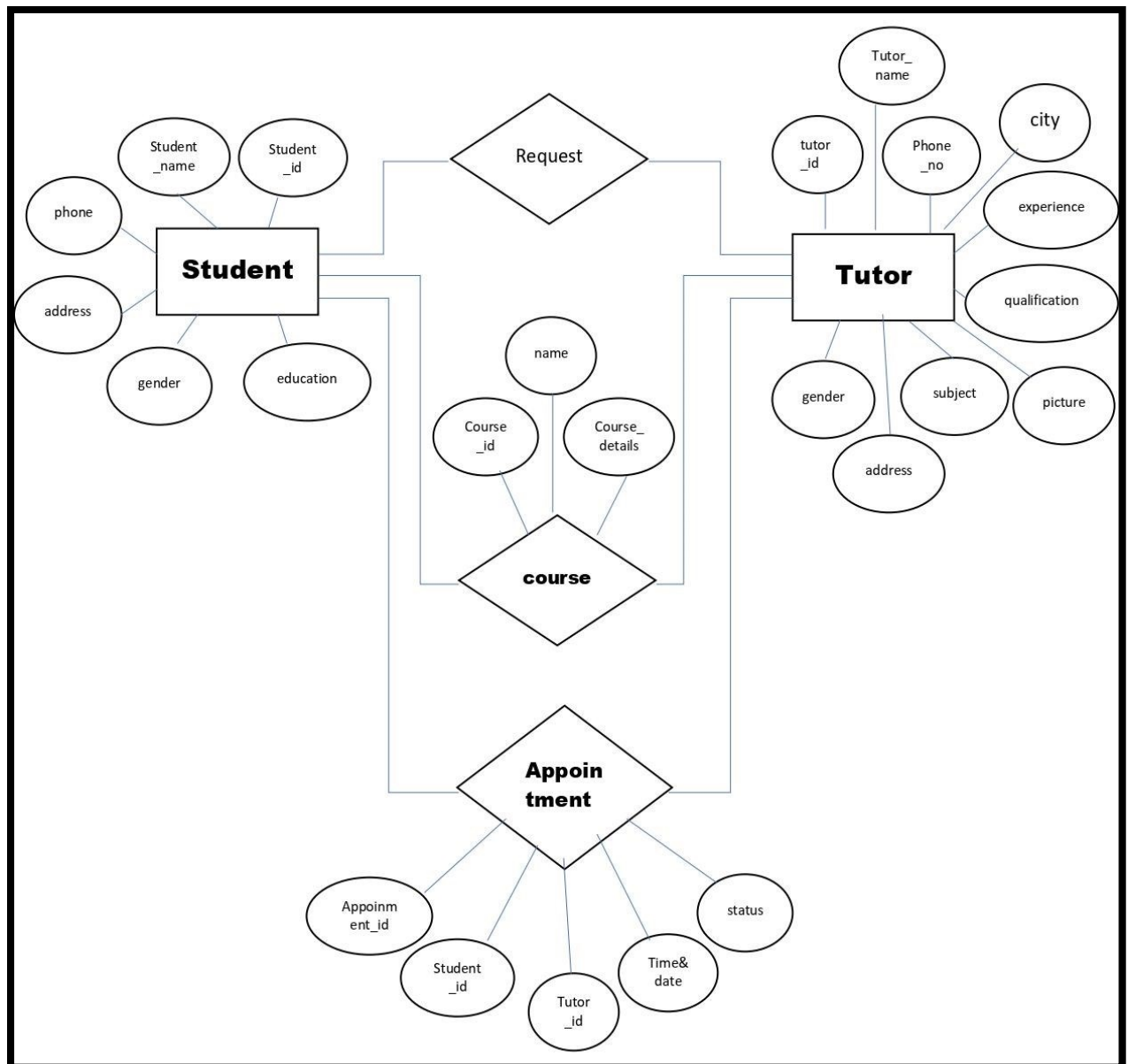
- Register: Students can register and obtain credentials.
- Login: Students can login using credentials.
- View Tutor: Students can check for tutors.
- Request a Demo Lecture: Students will select tutor and raise request for demo lecture.
- Book a Tutor: Students can book tutor.
- Rate Tutor: Students can rate tutor.
- View E Book: Students can check the E-books.

Chapter 3: System Analysis and Design

3.1 ER Diagram

An **Entity–Relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

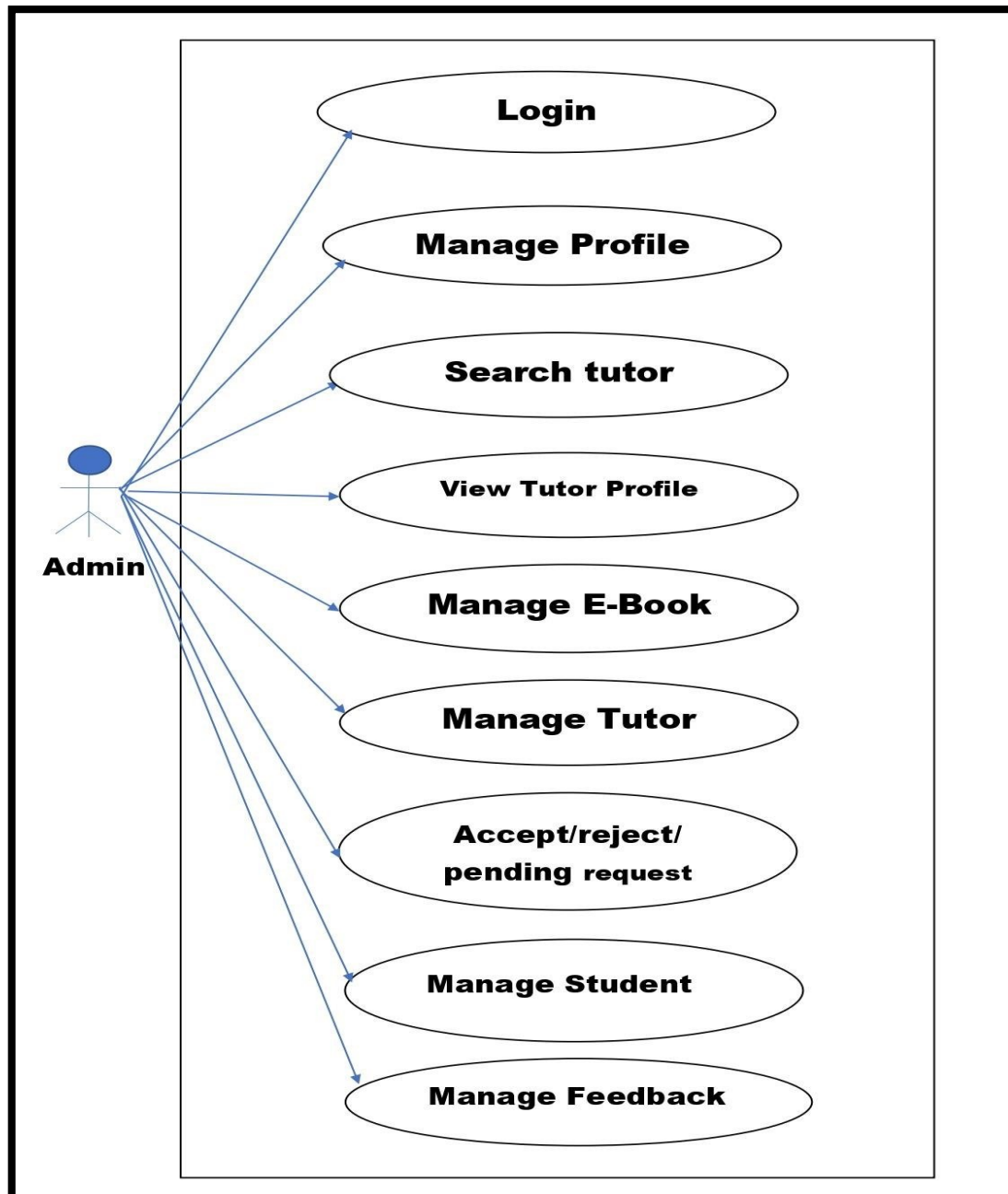
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.



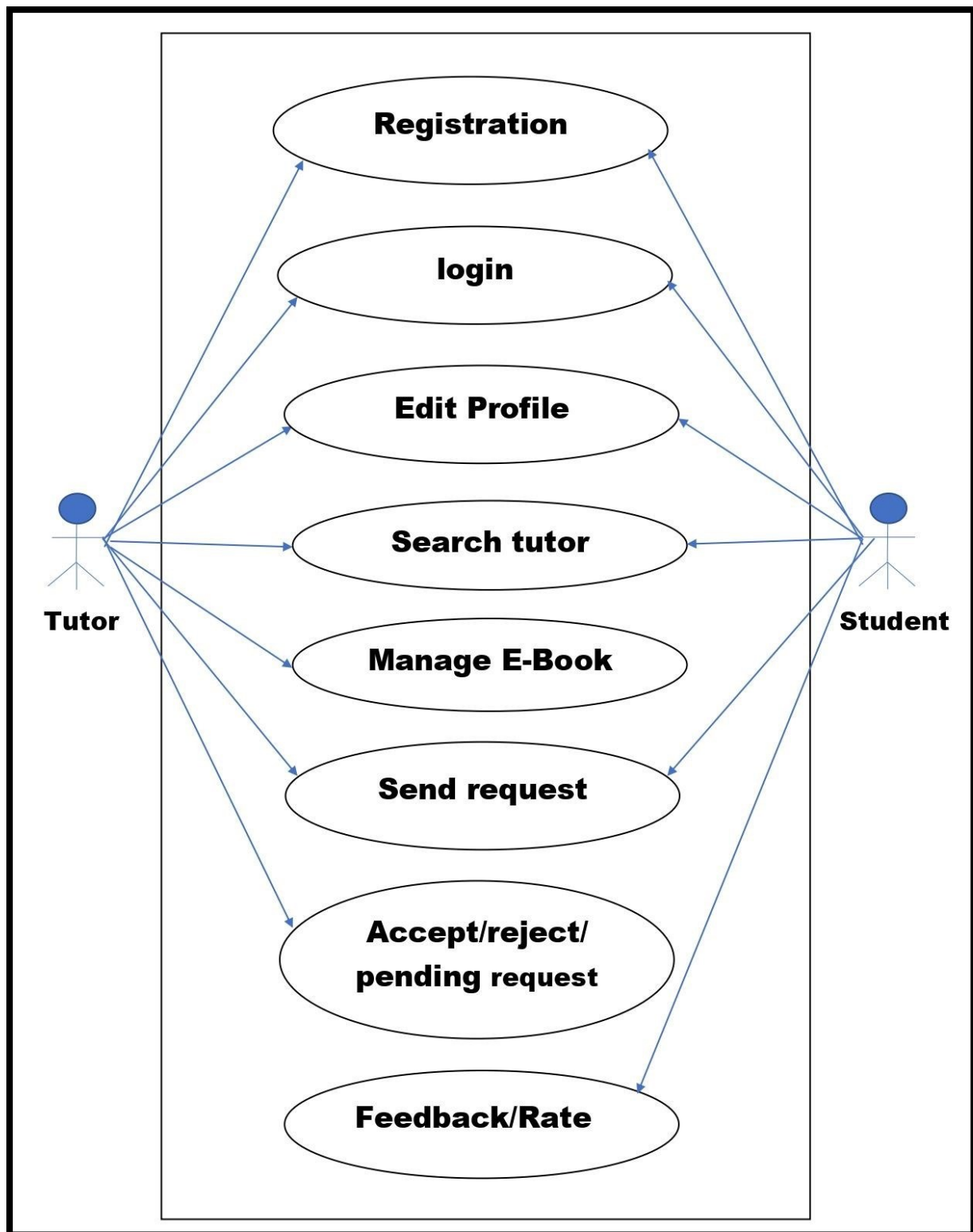
(Figure 3.1.1: ER Diagram)

3.2 Use Case Diagram

The use-case view is used to represent functionality of system which is connected with input and output user of the system. It divided the system functionality into transactions meaningful to actors. The use case view models the functionality of the system as perceived by users, called actors. A use case is coherent unit of functionality expressed as transaction among actors and the system. The purpose of the use case is to list the actors and use cases and show which actors participate in each use case. Use cases can also be described at various levels of detail they can be factored and described in terms of other, simpler use cases.



(Figure 3.2.1: Use Case Diagram)

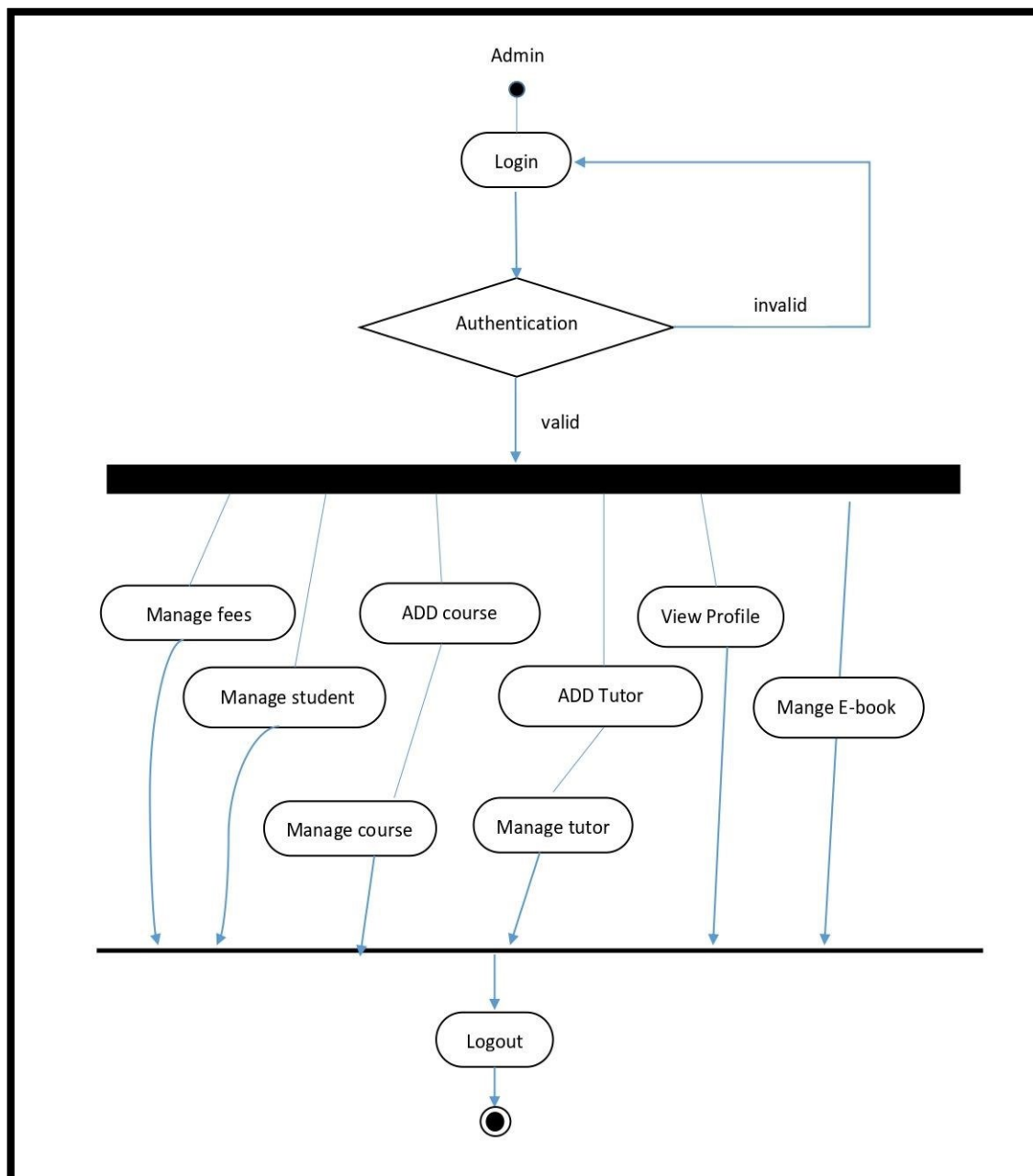


(Figure 3.2.2: Use Case Diagram)

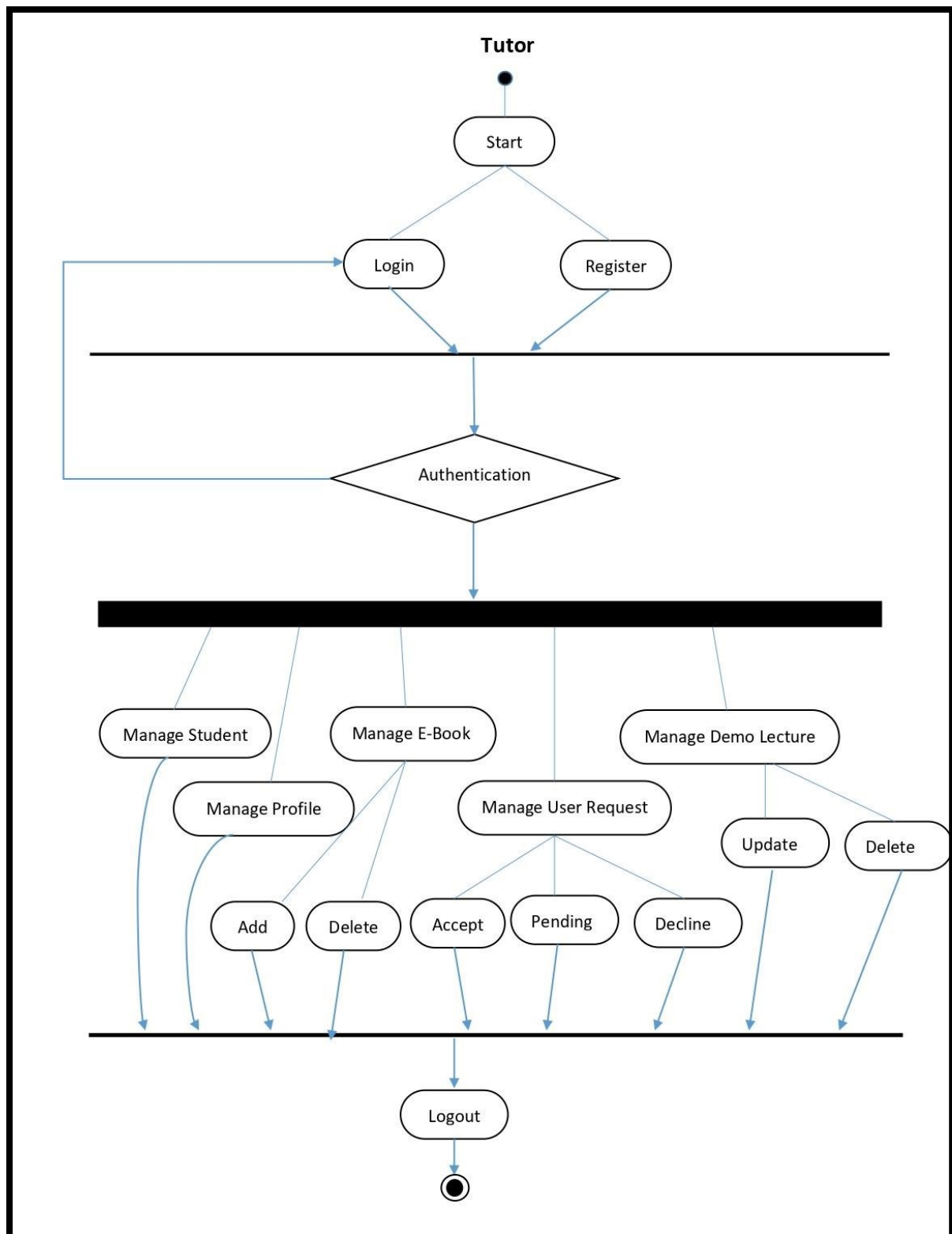
3.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

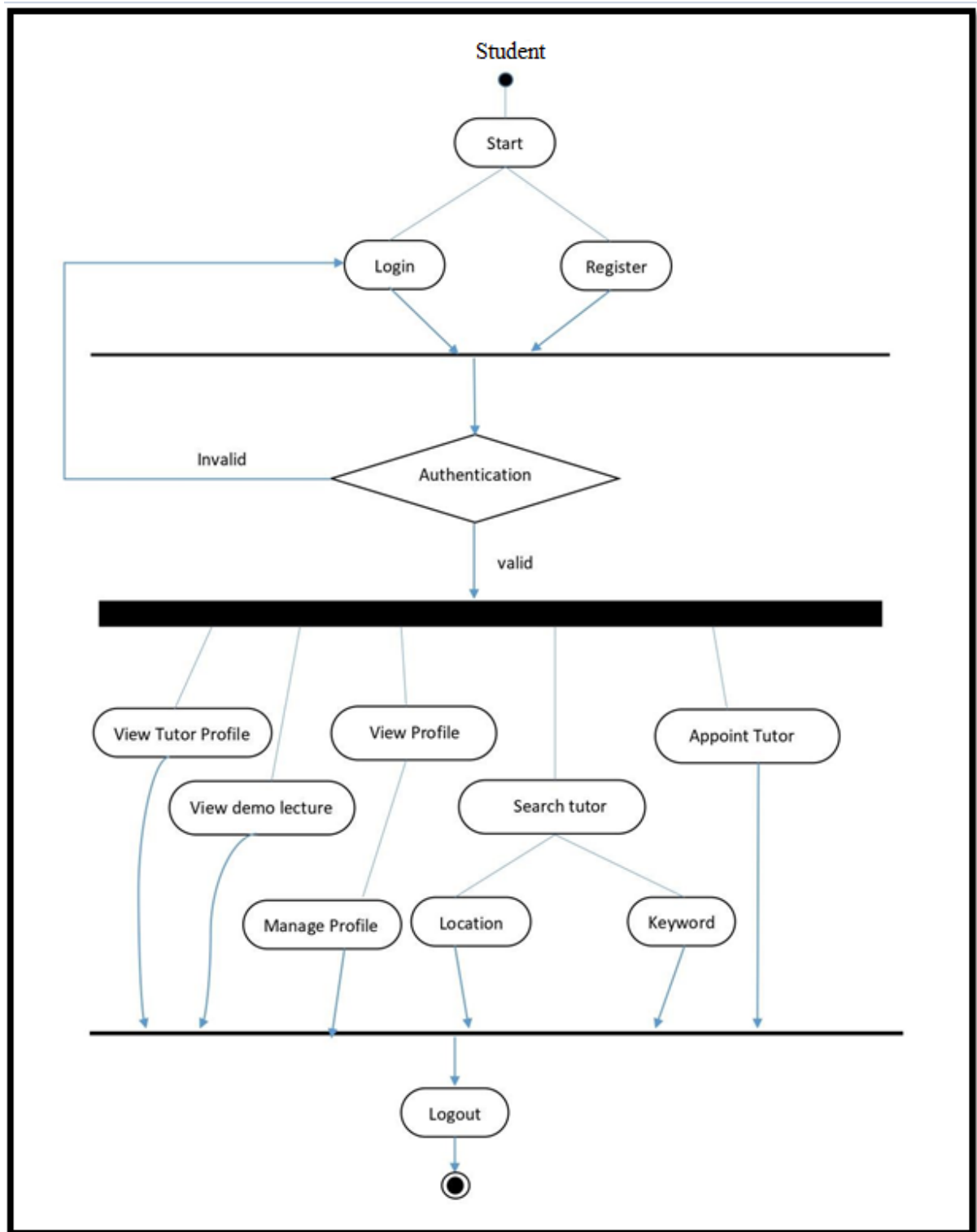
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.



(Figure 3.3.1: Activity Diagram for Admin)



(Figure 3.3.2: Activity Diagram for Tutor)

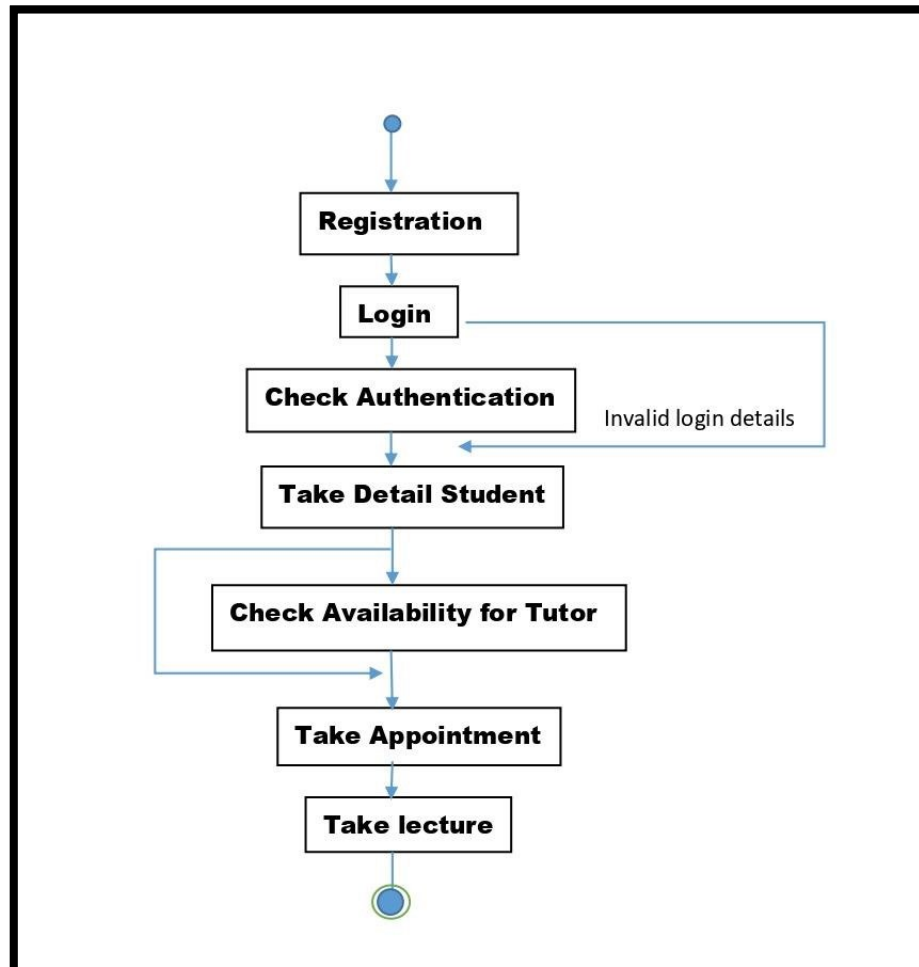


(Figure 3.3.3: Activity Diagram for Student)

3.4 State Diagram

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

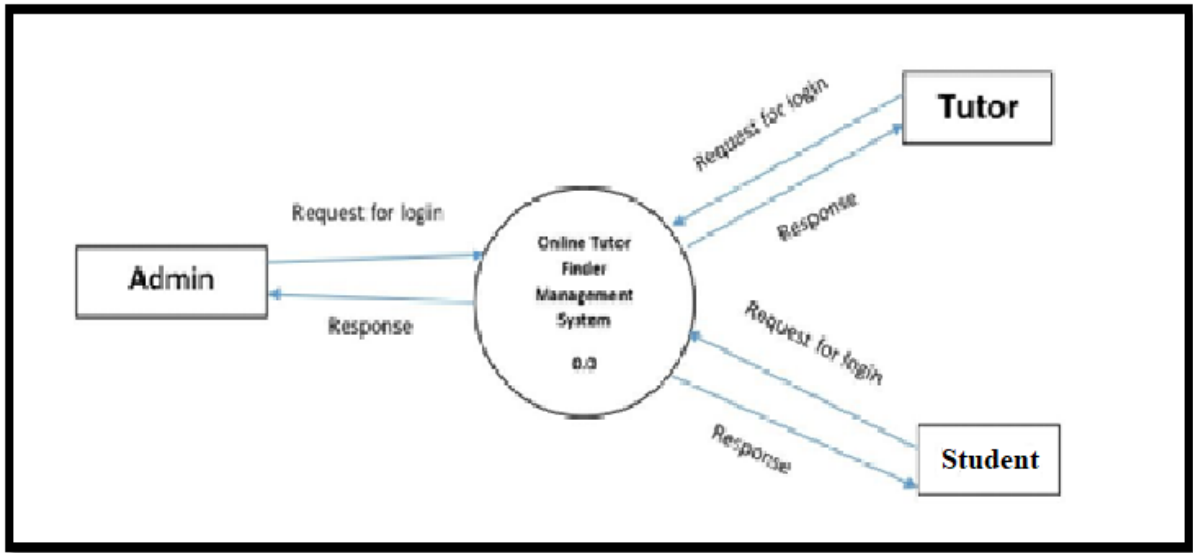
State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of state chart diagram is to model lifetime of an object from creation to termination.



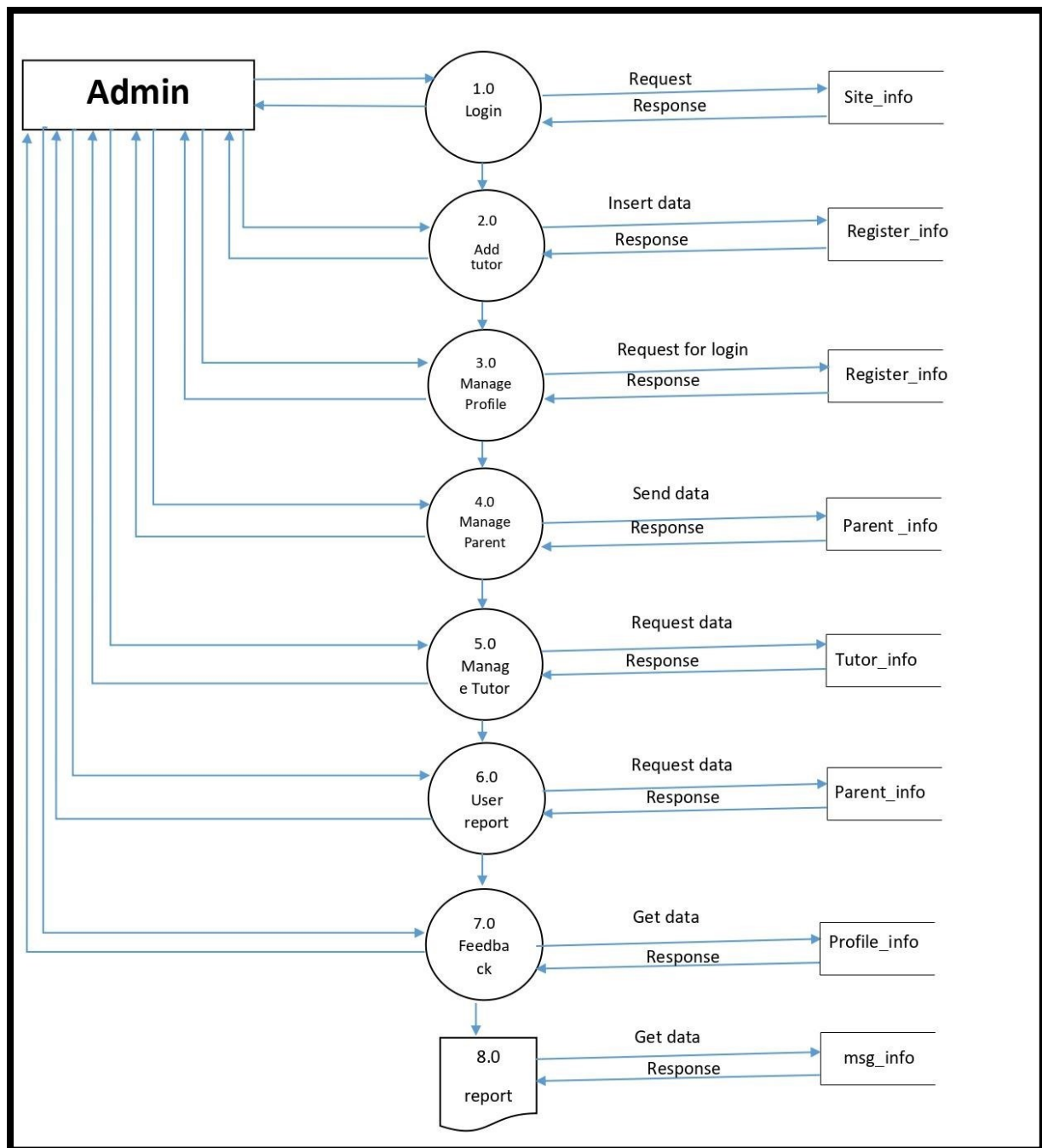
(Figure 3.4.1: State Diagram)

3.5 Data Flow Diagram

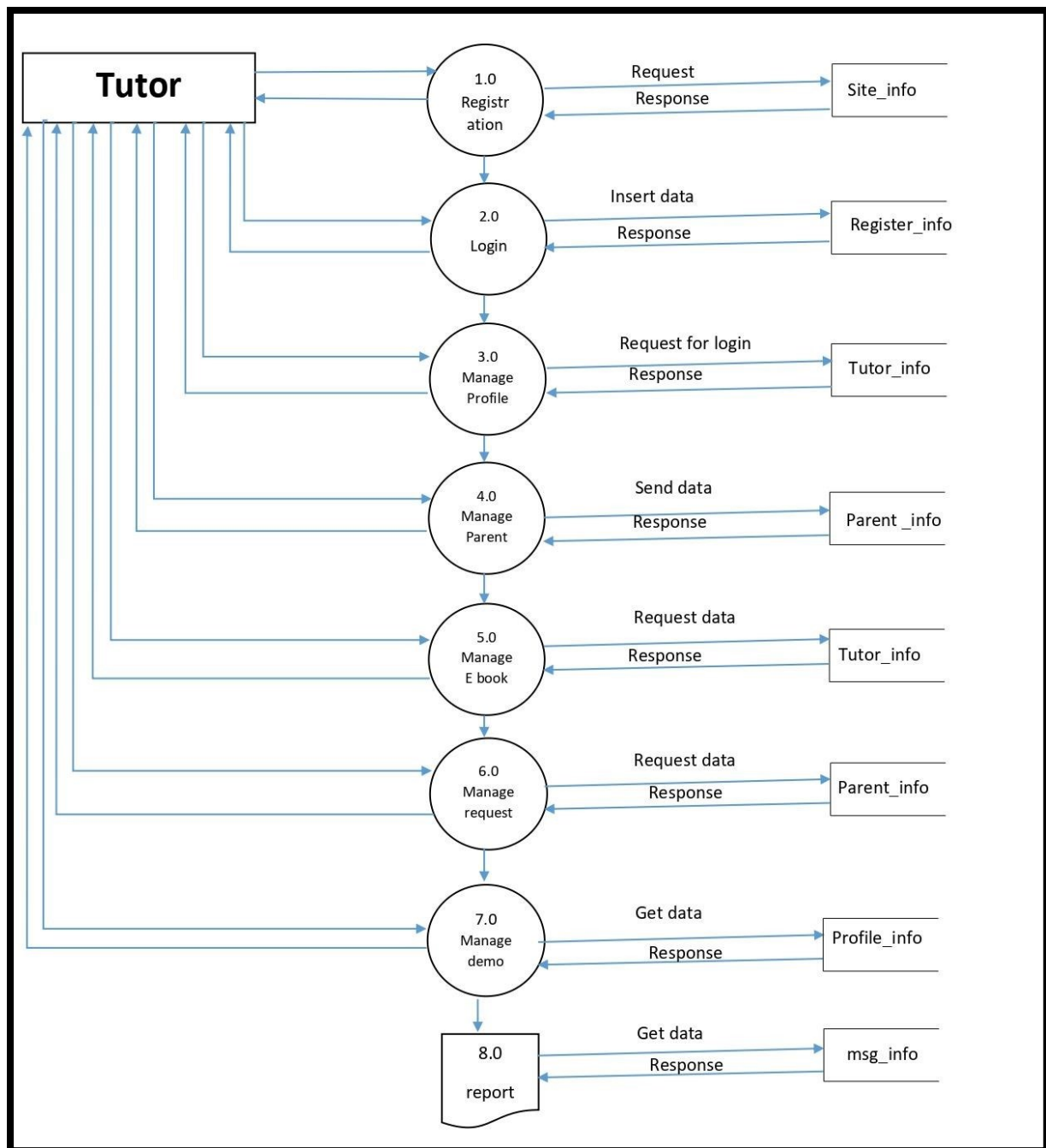
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.



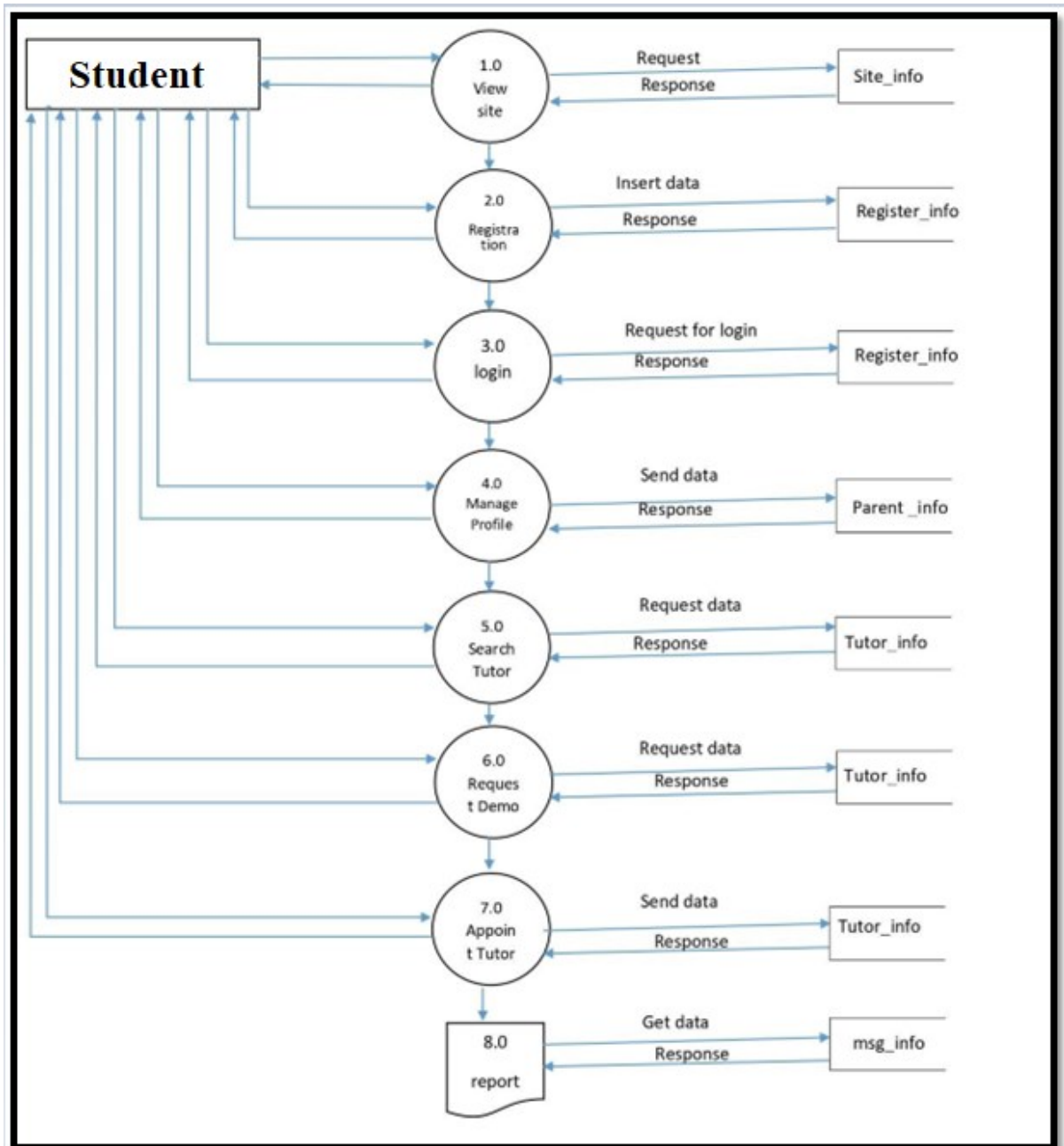
(Figure 3.5.1: DFD Level 0)



(Figure 3.5.2: DFD Level 1)



(Figure 3.5.3: DFD Level 2)



(Figure 3.5.4: DFD Level 3)

Chapter 4: Data Dictionary

Table Name: user

Description: This table shows the login details.

Field Name	Data Type	Constraint	Description
Id	Varchar(30)	Not Null	Primary Key
emailId	Varchar(35)	Not Null	Email for login
Password	Varchar(20)	Not Null	Password for login

Table Name: course

Description: This table shows the course details.

Field Name	Data Type	Constraint	Description
id	Varchar(30)	Not Null	Primary Key
name	Varchar(35)	Not Null	Name of course
image	Varbinary(max)	Not Null	Image of course

Table Name: Appointment

Description: This table shows the appointment details.

Field Name	Data Type	Constraint	Description
id	Varchar(30)	Not Null	Primary key
sId	Varchar(30)	Not Null	Id of student
tId	Varchar(30)	Not Null	Foreign key
created	Time	Not Null	Created information
updated	Time	Not Null	Updated information
deleted	Time	Not Null	Deleted information