# **Design**

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

## Structural Design

### **1.1 Final Class Diagram**

### **1.2 Flow Chart**

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step.

### **Justification**

Various reasons are there to design flowchart. Some of them are listed below:

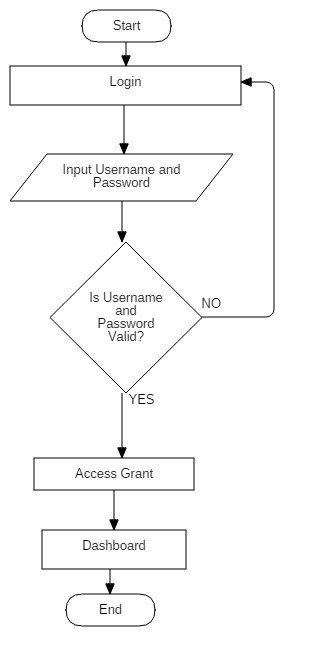
* It shows how a process is performed from start to finish, typically in sequential order Troubleshooting of the logical problems are easy while designing the ER model.
* It helps to identify how long it takes to complete each task in a specific process.

It is difficult to fix bugs and problems in workflow, if you do not know what went wrong and at which stage in the process it went wrong.

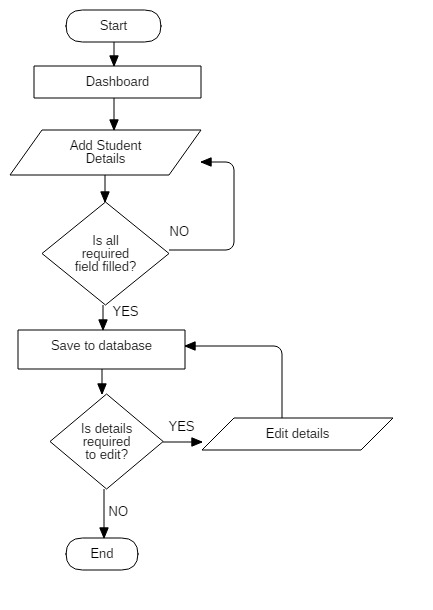
### **Notation used**

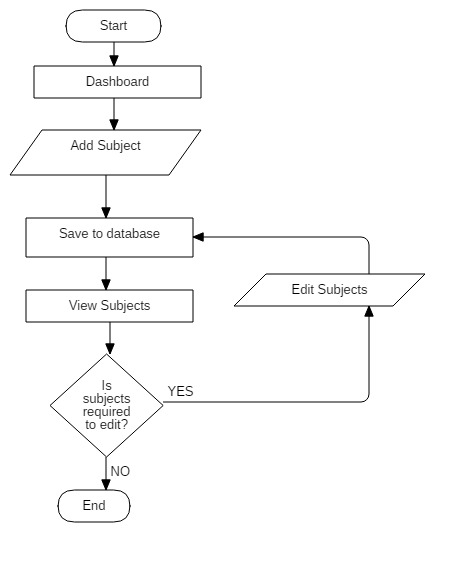
### **Actual Flow Chart:**

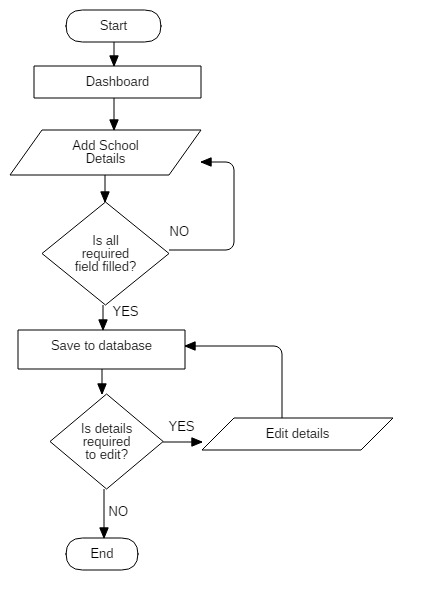
The Flow chart for the Mark Sheet Generator is shown below:

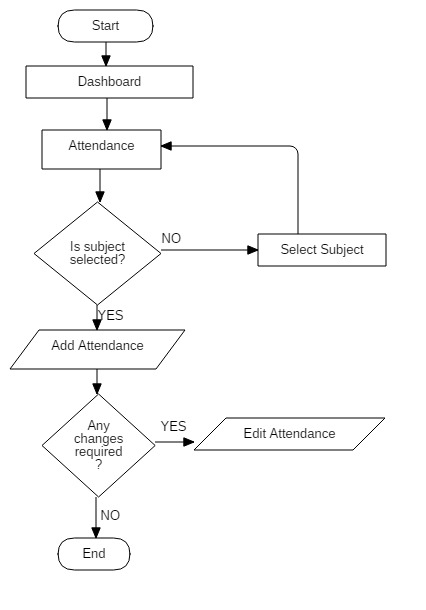


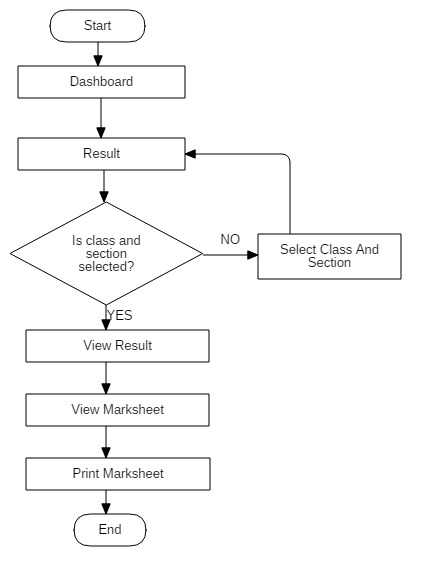
The aside flowchart shows login procedure. Here, login into the account is after verifying the valid username and password of the user.

 The aside flowchart is of inserting the student details. Here, adding student detail is inserted into the database only after making sure, if the entire required filled are provided. If any changes are to be made then changes can be made.

 The aside flowchart is of adding the subjects. We can add subjects. After adding the subjects, if changes are to be made then it is possible.

Aside flowchart shows flow of adding school information. We can add school details. If all the information is not filled then it is to be filled before saving. Then any changes are to be made is possible. 

Aside is flowchart of adding attendance of students. To make attendance subject and class is to be selected. Any edit can be possible to make. 

Aside is flowchart of view result. Result can be view only after providing class and section. After that user can view mark sheet of the students. Mark sheet can also be download in pdf format.

## **Behavioral Diagram**

### **2.1 Activity Diagram**

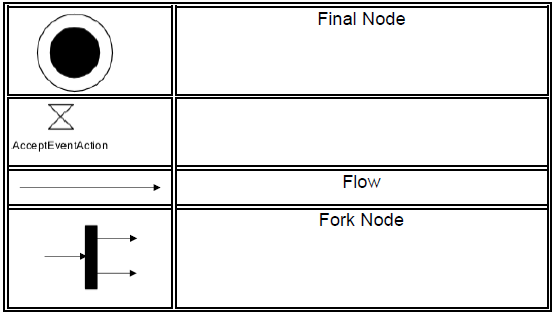
Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

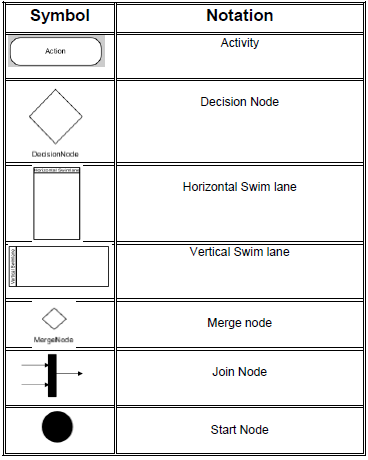
### **Justification**

Various reasons are there to design Activity diagram. Some of them are listed below:

* Helps to understand the workflow of the activity of the system.
* Helps to describe the complicated sequential algorithm.
* Helps to deal with multithreaded application.
* Helps to analyze use cases of the system.

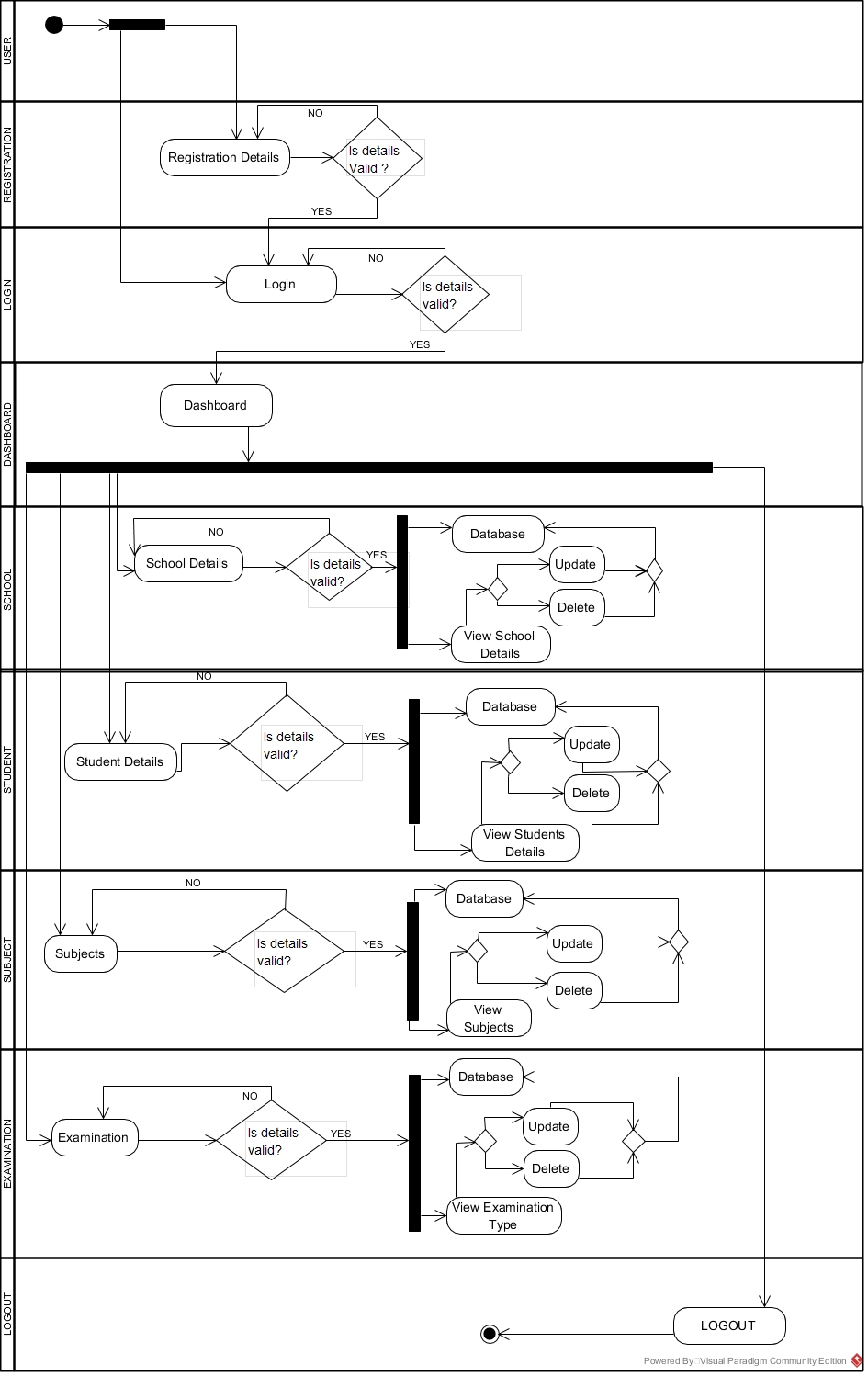
### **Notation used**



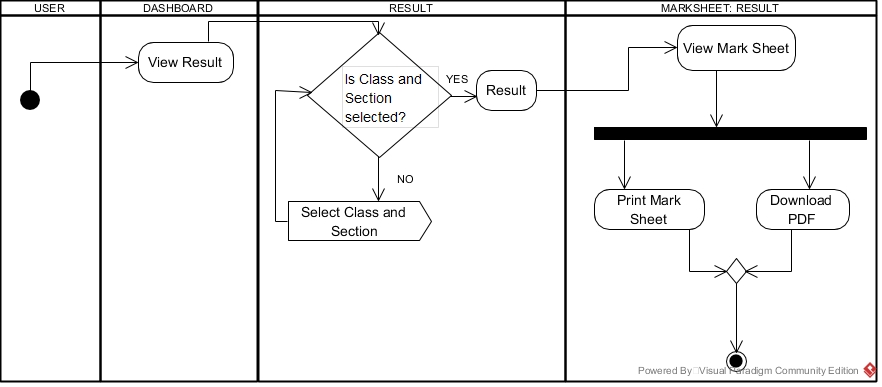


### **Actual Activity Diagram :**

The Activity Diagram for the Mark Sheet Generator is shown below:

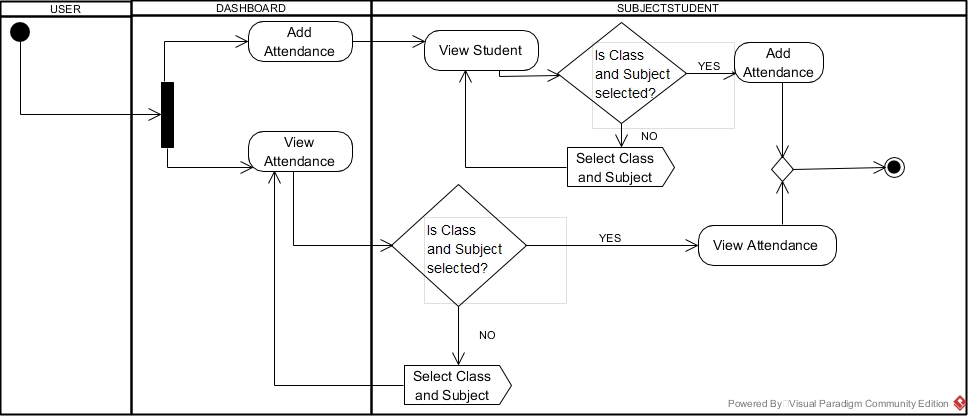
The activity diagram shows the flow of the activity of the system. In Mark Sheet Generator, initially user is logged in to the system only after the user credentials are verified. Before login user have to get registration if user is not registered. After logged into the system dashboard is displayed. User can perform multiple functions of the system.

Firstly, if user have to set the school information, user can navigate to school details. Here, user provide school information and unless it is valid and compulsory filled are entered, provided data will not be store in the database. To edit the details user can view and edit the details.

Likewise, the procedure are same for storing student details, subjects and examination type. User can sing out the system as well.

The above activity diagram shows the activity for result. After user completes the procedure for marks obtained, user can view result. The result is generated automatically after storing the marks obtained. To view result, user navigates from dashboard to view result. Then it is ensure that class and section is selected. Result can be view and also user can view mark sheet. User can either download mark sheet in pdf or can print from their local printer.

If class and section not provided then selection for these fields message is being flowed to user.



The above activity diagram represents the activity for attendance of the students.

Initially user navigates to dashboard where user gets two option that can be perform simultaneously. If user wishes to add the attendance, user required to view the students. After that it is make sure that class, section and subject are selected or not. If it true then only attendance can perform. Else a message is send to user to identify the class, section and subject.

To view the attendance, it is ensure that class and subject is selected. Then attendance can be view.

### **2.2 Sequence Diagram**

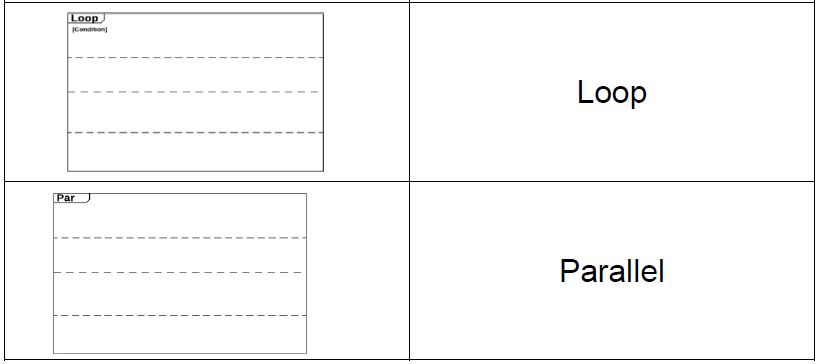
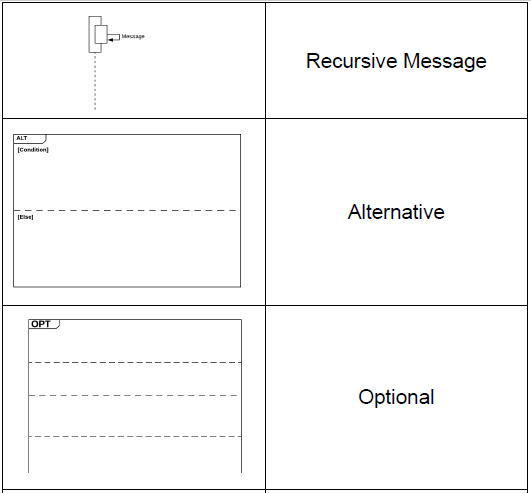
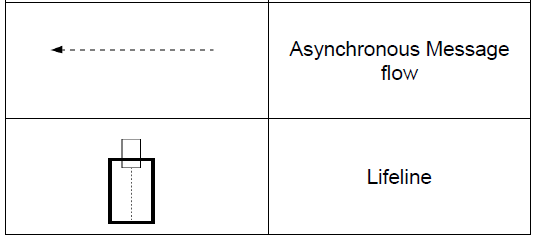
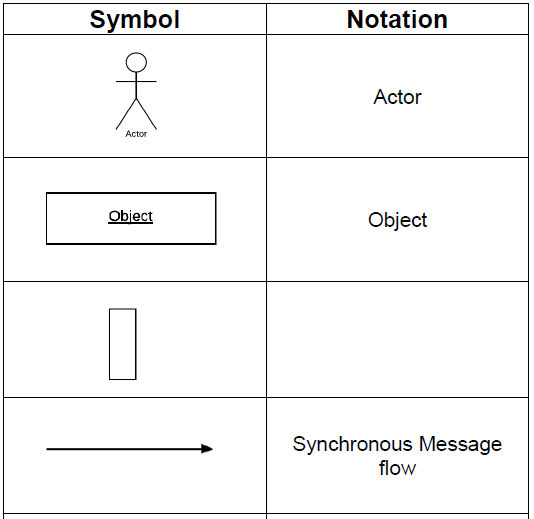
Sequence Diagram is the behavioral diagram which prime motive is to represent the message flow between the objects of the system. It shows different types of message flow and lifelines of the messages and objects in the system.

### **Justification**

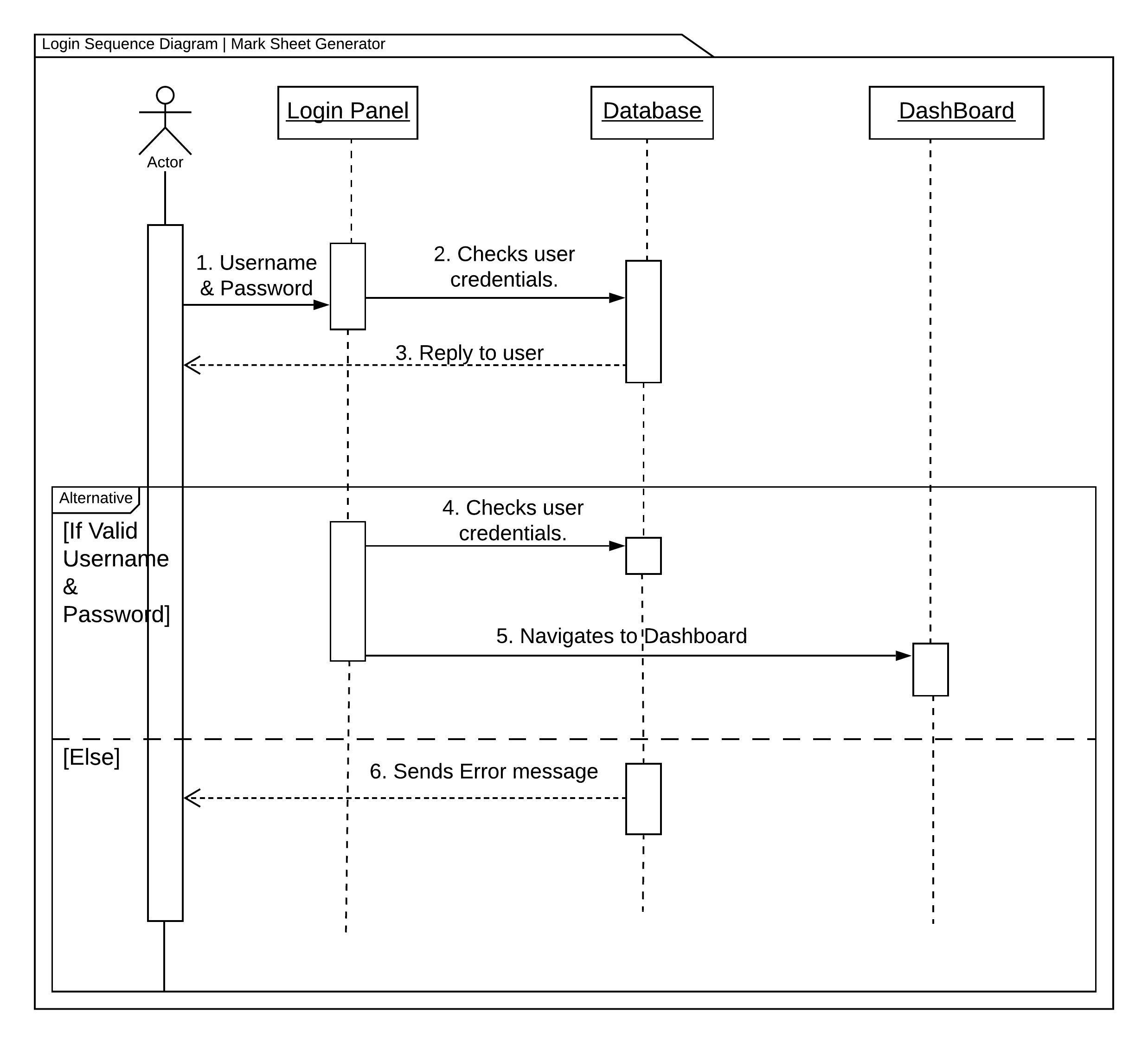
Reasons of creating sequence diagram are:

* Sequence Diagram is drawn to show the behavior of objects contained by a single use case.
* It is good at showing the association among the objects.
* It is drawn to get to know about the thorough functionality of systems.
* In order to visualize the logic behind function sequence diagram is drawn.

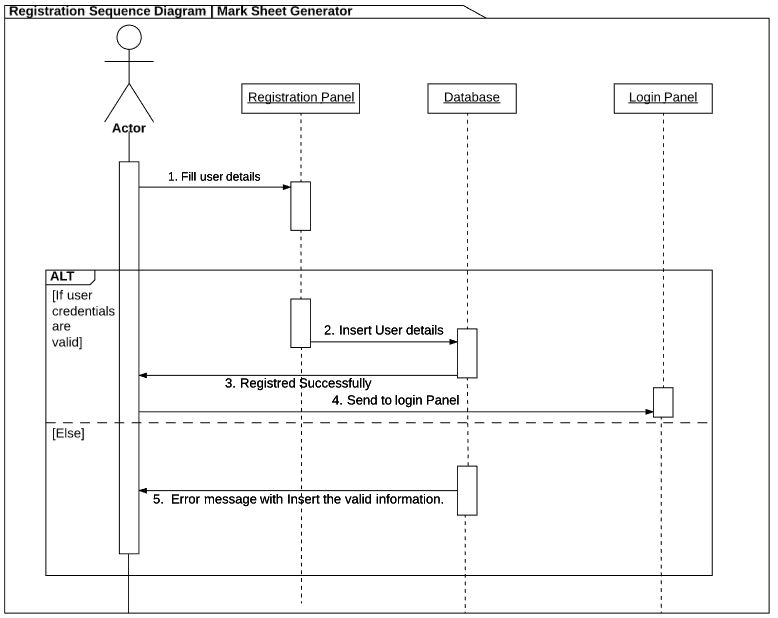
### **Notation**



### **Actual Sequence Diagram**



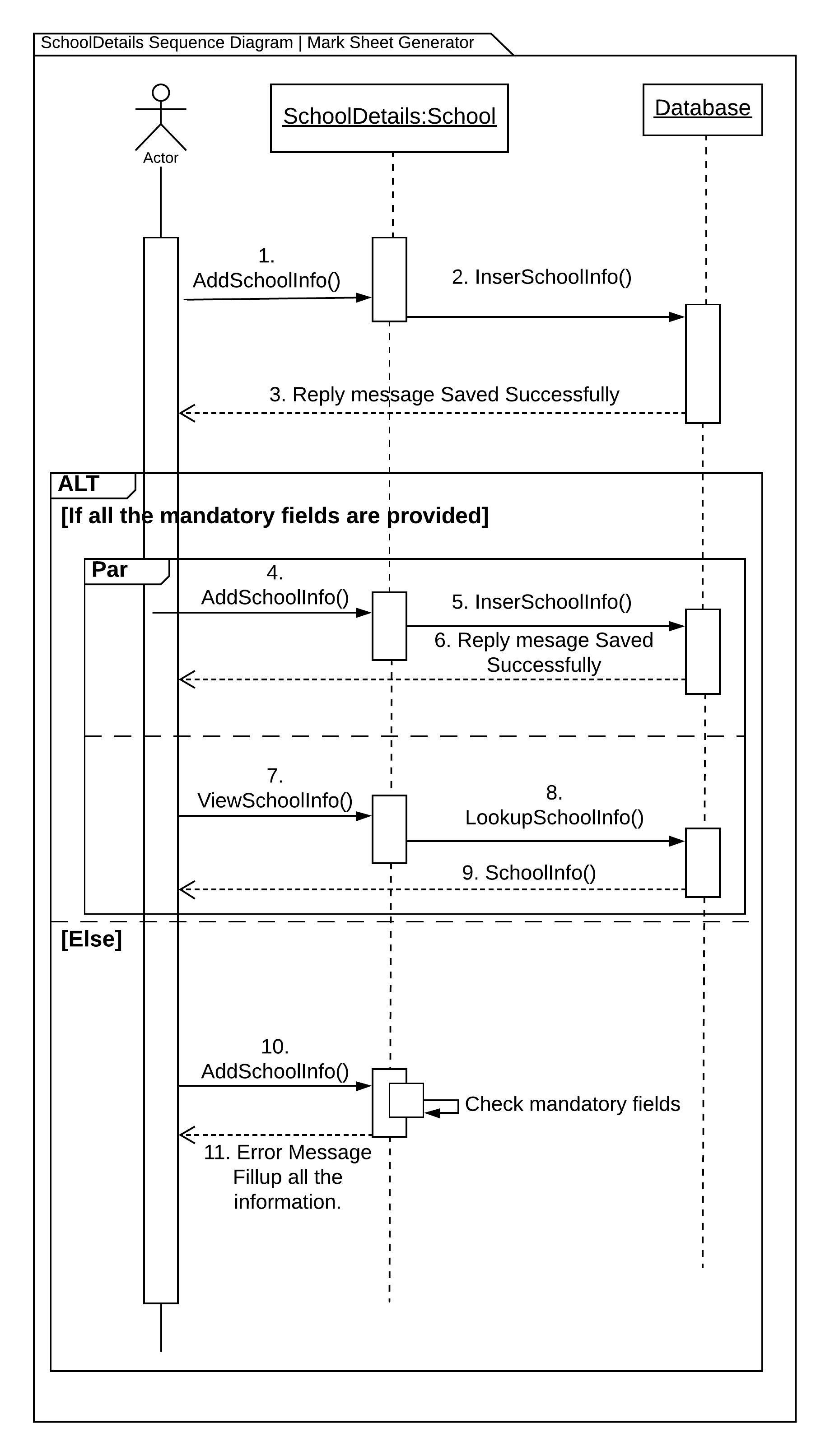
The above sequence diagram is the sequence diagram for login. Here the message flow from login panel to the database system is shown. The alternative is used to show the flow of message when the credentials are true and not true.

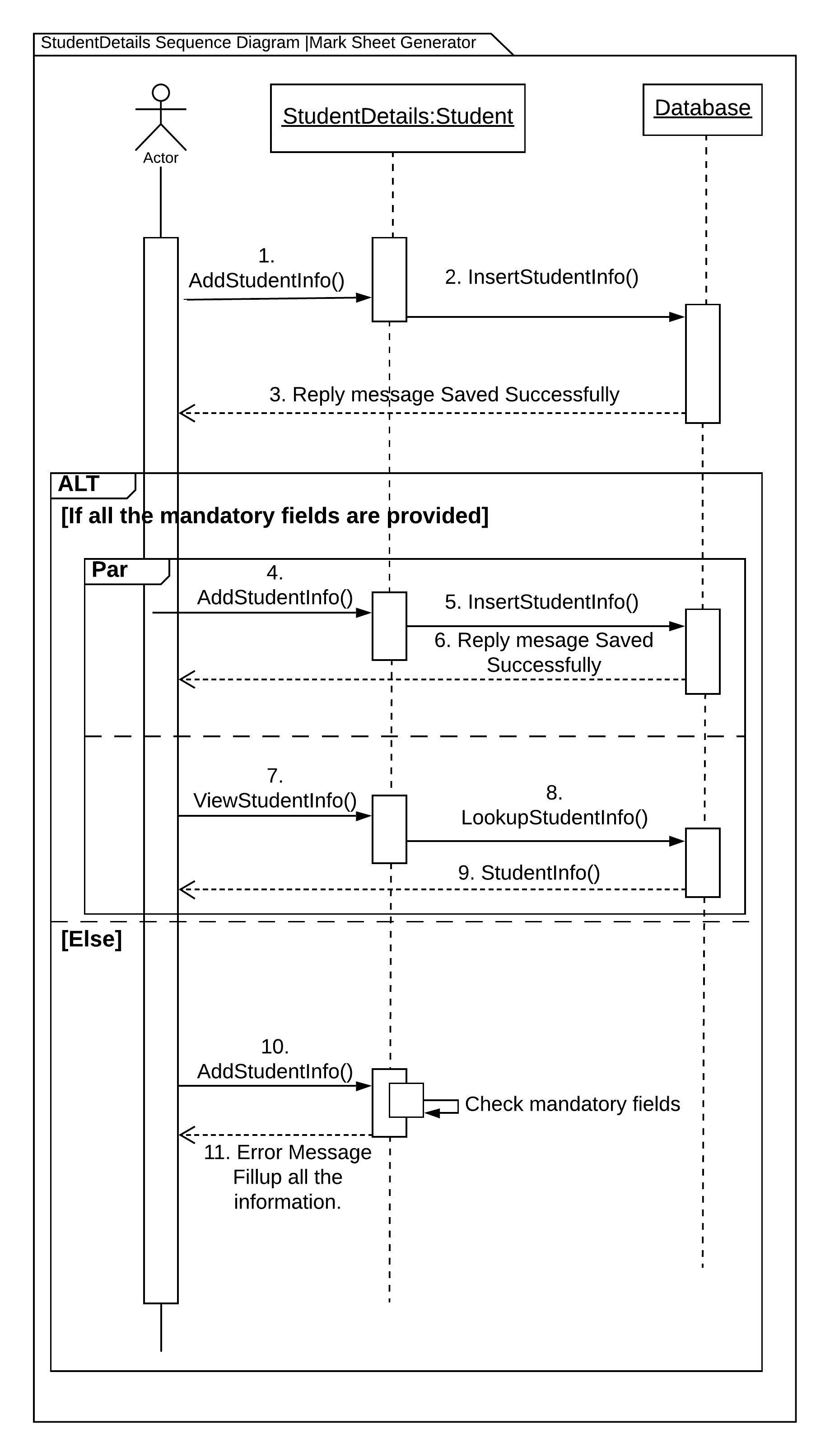


Above is the sequence diagram of the registration. Here user message flow of registration credentials with database system is illustrated. The flow of message is shown in alternative sequence diagram. If the user provides valid information (appropriate data with the correct or assigned datatype) then message is let to flow to login panel for user to be logged in to the system. If any criteria does not meet up then error message is being forwarded.

The following sequence diagram is showing the sequence of the message flowing while school details are inserting into the database.

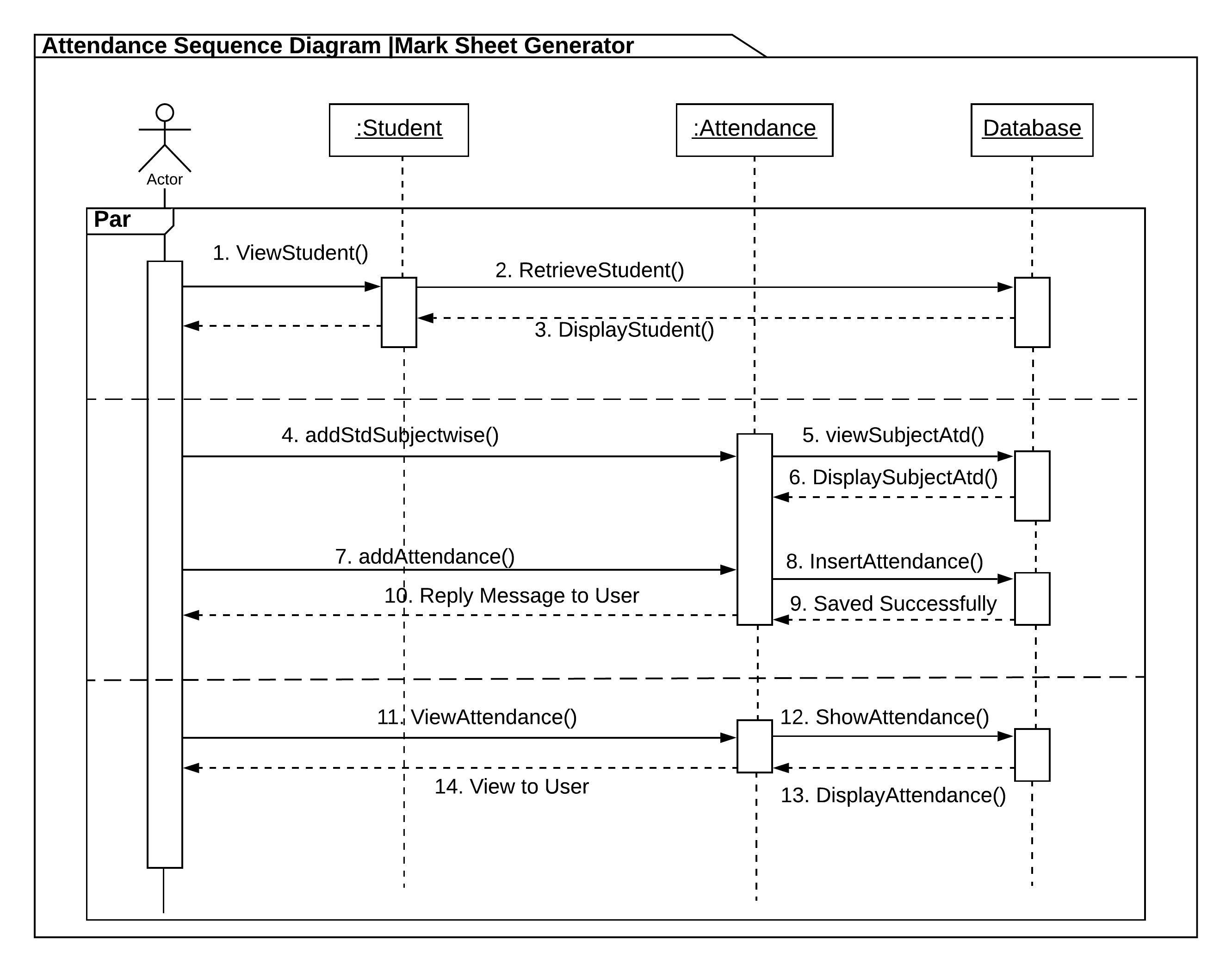
Here I have integrated alternative and parallel sequence. In Parallel case there are inserting and retrieval school information into the database. While in alternative if all the compulsion fields are entered then the entire parallel are allowed else it will send error message.





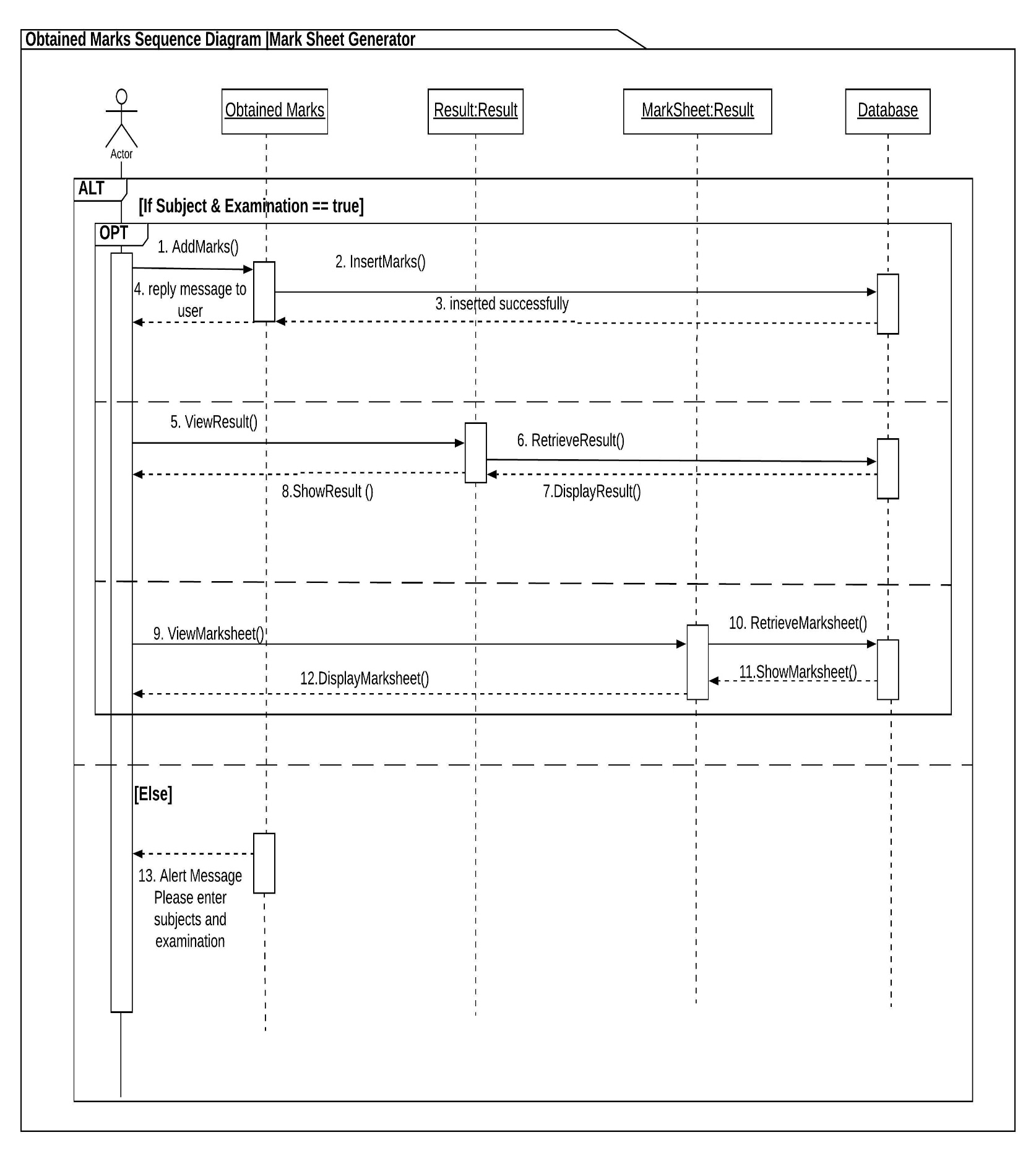
The following sequence diagram is showing the sequence of the message flowing while student details are inserting into the database.

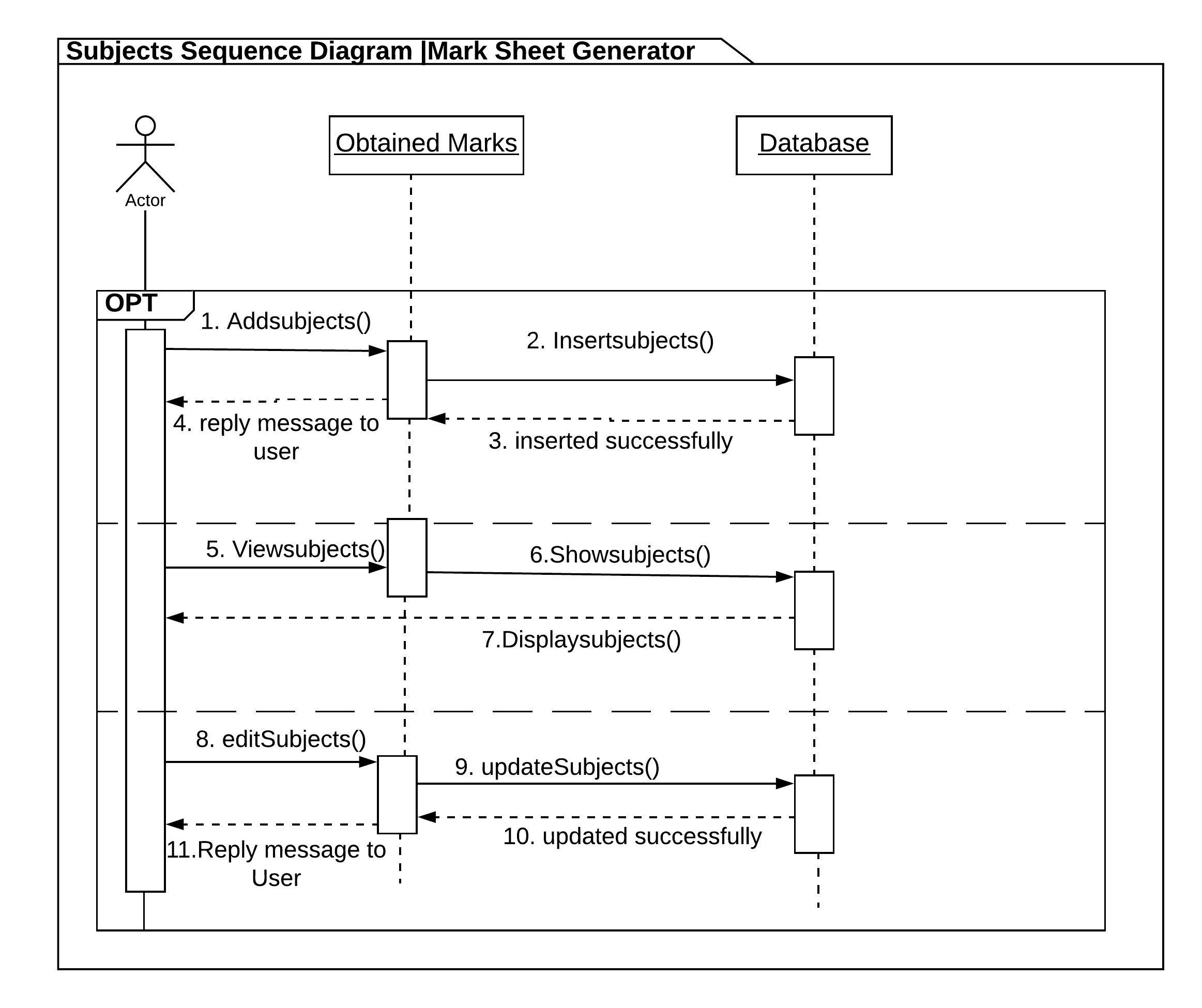
Here the integration of alternative and parallel sequence. In Parallel case there are inserting and retrieval students information into the database. While in alternative if all the compulsion fields are entered then the entire parallel are allowed else it will send error message.



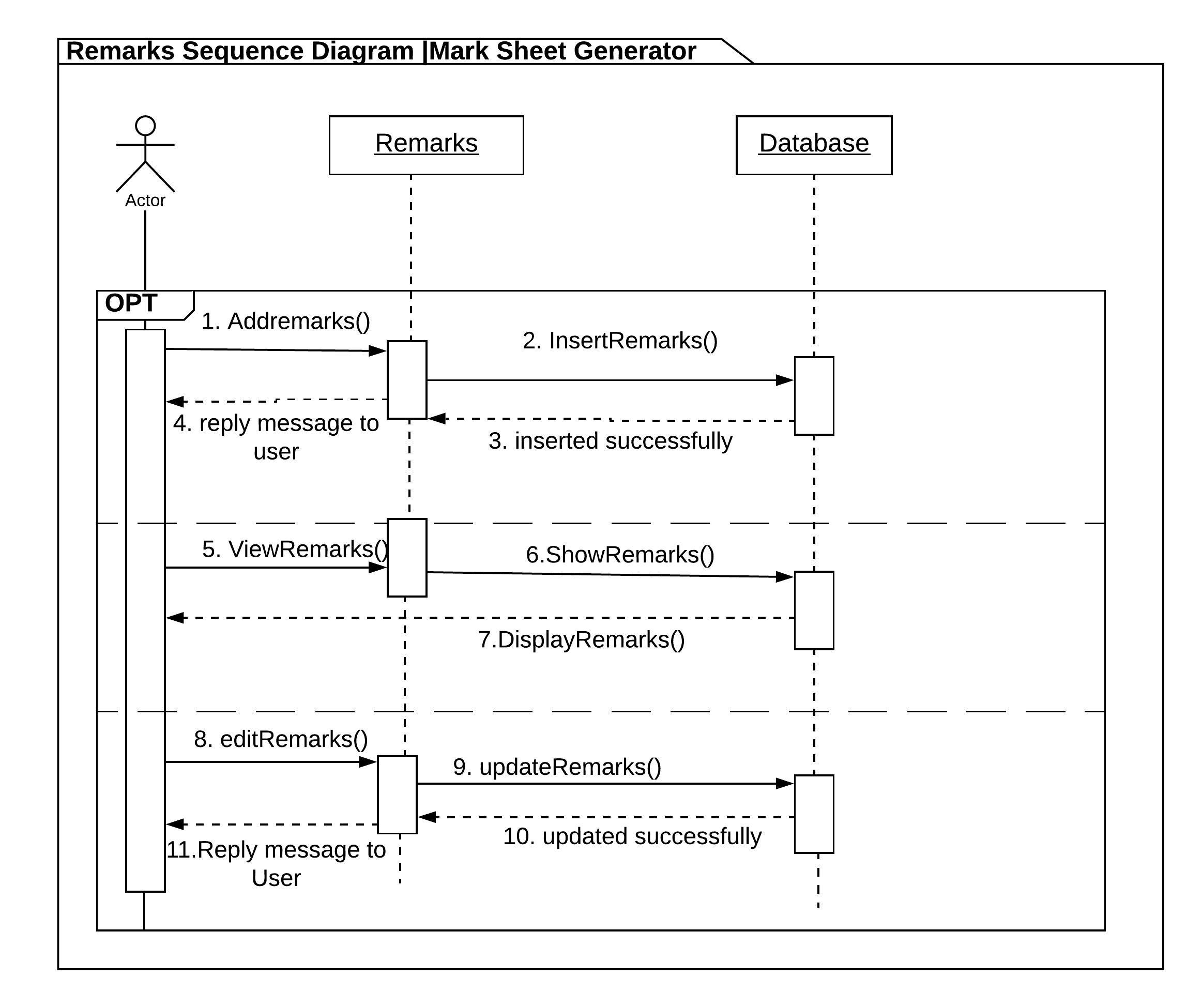
Attendance sequence is in parallel sequence. The message flow between different objects, classes and database are shown in the sequence diagram.

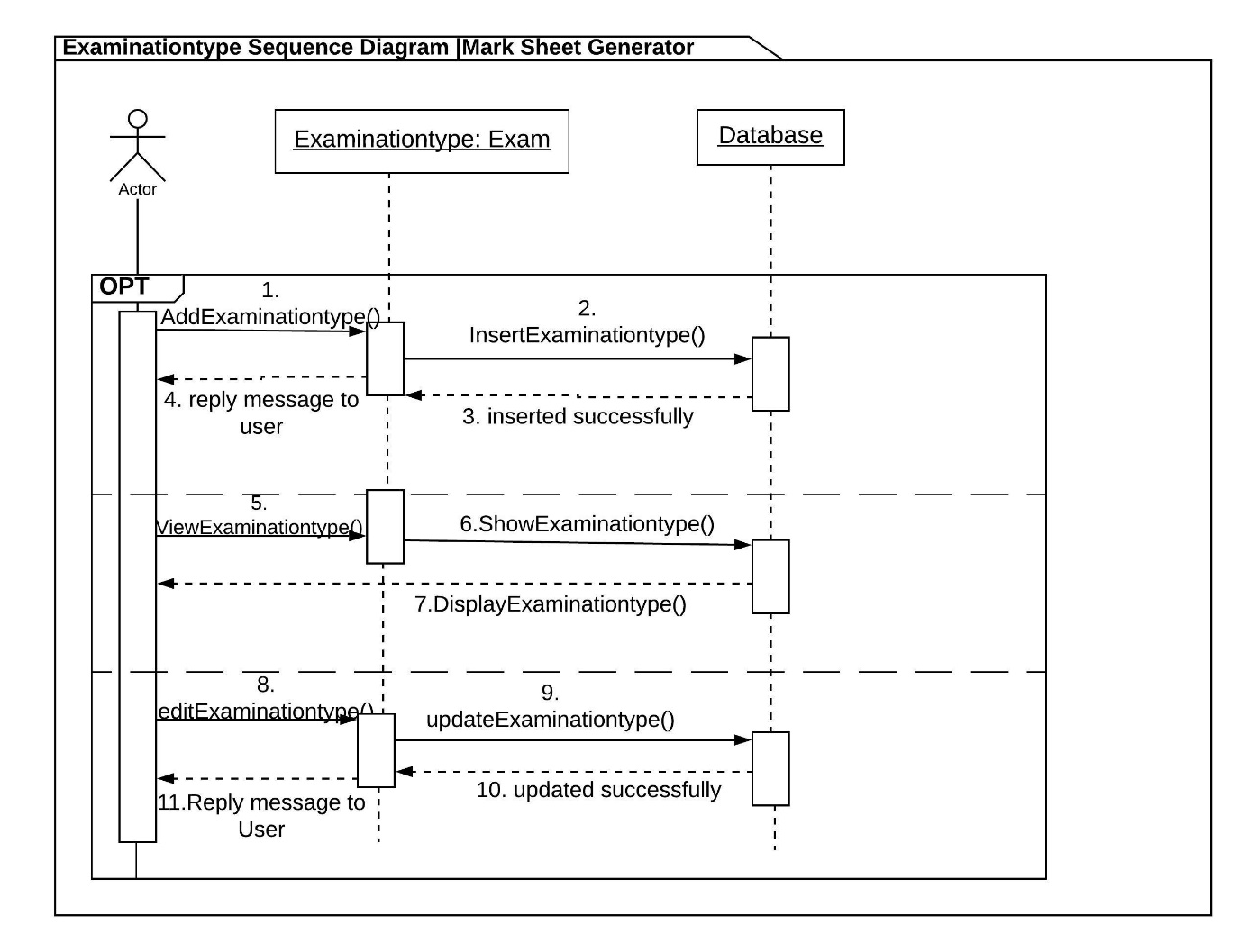
Obtained Marks sequence is in parallel sequence. The message flow between different objects, classes and database are shown in the sequence diagram.





The Subject sequence diagram is shown in optional sequence. Here message flow between objects obtained marks and database are shown.

The remarks sequence diagram is shown in optional sequence. Here message flow between objects remarks and database are shown.



The examination sequence diagram is shown in optional sequence. Here message flow between examination and database are shown.

## **Database Model**

### **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **School** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| school\_Id | Integer |  | Primary Key | Not Null |
| school\_Name | Varchar | 255 |  | Null |
| address | Varchar | 255 |  | Null |
| email | Varchar | 255 |  | Null |
| phone | Varchar | 255 |  | Null |
| established\_Date | date |  |  | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| student\_Id | Integer |  | Primary Key | Not Null |
| full\_Name | Varchar | 255 |  | Null |
| class | Varchar | 255 |  | Null |
| section | Varchar | 255 |  | Null |
| roll\_No | Varchar | 255 |  | Null |
| gender | Varchar | 255 |  | Null |
| date\_of\_Birth | Date |  |  | Null |
| address | Varchar | 255 |  | Null |
| parents\_Name | Varchar | 255 |  | Null |
| phone | Varchar | 255 |  | Null |
| photo | Varchar | 255 |  | Null |
| attendanceRecord\_Id | Integer |  | Foreign Key | Null |
| school\_Id | Integer |  | Foreign Key | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| subject\_Id | Integer |  | Primary Key | Not Null |
| subject | Varchar | 255 |  | Null |
| full\_Marks | Integer |  |  | Null |
| pass\_Marks | Integer |  |  | Null |
| marks\_Obtained | Float |  |  | Null |
| examination\_Id | Integer |  | Foreign Key | Null |
| attendanceRecord\_Id | Integer |  | Foreign Key | Null |
| result\_Id | Integer |  | Foreign Key | Null |
| examination | Varchar | 255 |  | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Examination** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| examination\_Id | Integer |  | Primary Key | Not Null |
| examination\_Type | Varchar | 255 |  | Null |
| **Remark** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| remarks\_Id | Integer |  | Primary Key | Not Null |
| remarks | Varchar | 255 |  | Null |
| result\_Id | Integer |  | Foreign Key | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| class\_Id | Integer |  | Primary Key | Not Null |
| class | Integer |  |  | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| section\_Id | Integer |  | Primary Key | Not Null |
| section | Varchar | 255 |  | Null |
| class\_Id | Integer |  | Foreign Key | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attendance** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| atd\_Id | Integer |  | Primary Key | Not Null |
| total\_Workingdays | Integer |  |  | Null |
| present\_Days | Integer |  |  | Null |
| absent\_Days | Integer |  |  | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AttendanceRecord** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| atdRecord\_Id | Integer |  | Primary Key | Not Null |
| subject\_Name | Varchar | 255 |  | Null |
| student\_Name | Varchar | 255 |  | Null |
| attendance | Integer |  |  | Null |
| class | Integer |  |  | Null |
| section | Varchar | 255 |  | Null |
| total\_workingdays | Integer |  |  | Null |
| present\_Days | Integer |  |  | Null |
| absent\_Days | Integer |  |  | Null |
| attendance\_Id | Integer |  | Foreign Key | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Result** | | | | |
| **Name** | **Data Type** | **Length** | **Key** | **Null** |
| result\_Id | Integer |  | Primary Key | Not Null |
| student\_Name | Varchar | 255 |  | Null |
| class | Integer |  |  | Null |
| roll\_No | Integer |  |  | Null |
| section | Integer |  |  | Null |
| subject | Varchar | 255 |  | Null |
| examination\_Type | Varchar | 255 |  | Null |
| attendance | Integer |  |  | Null |
| remark | Varchar | 255 |  | Null |
| marks\_Obtained | Integer |  |  | Null |
| division | Varchar | 255 |  | Null |
| examination\_Id | Integer |  | Foreign Key | Null |
| class\_ID | Integer |  | Foreign Key | Null |
| section\_ID | Integer |  | Foreign Key | Null |

### **3.2 ER Model**

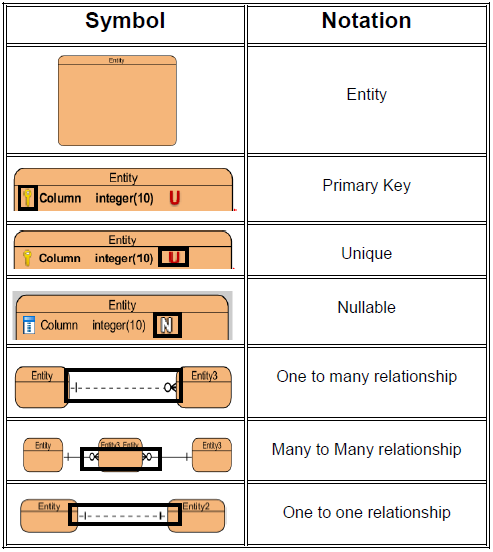
ER Diagram is the diagram that represents the logical structure of database. It shows all the possible relationship between the entities. It consists of entities and attributes of entities in the relationships.

### **Justification**

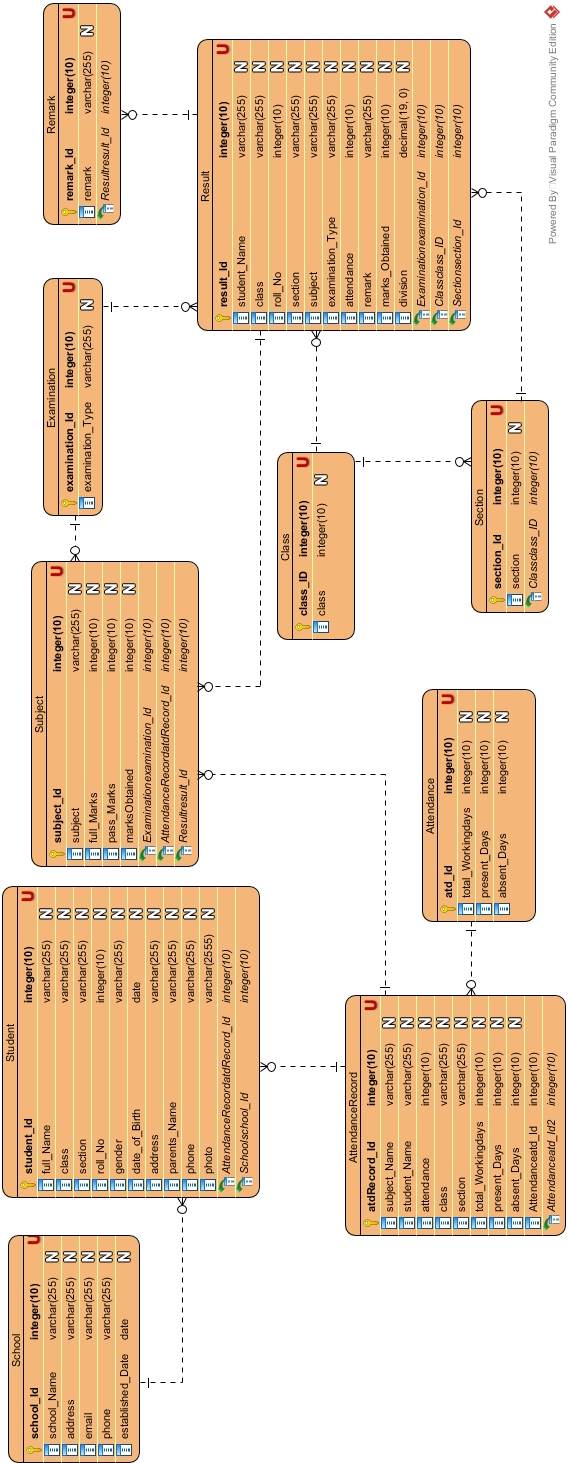
Various reasons are there to design ER diagram. Some of them are listed below:

* In order to determine the requirements of the system, ER model is designed in physical data modelling phase.
* Troubleshooting of the logical problems are easy while designing the ER model.
* ER model can be the base to create database of the system and conserve the time in designing of the databases.
* ER model can help in analyzing about data.

### **Notation used**



### **Actual ER Diagram:**

The ER model for the Mark Sheet Generator is shown below:

In the ER model above, I have created entities like School, Student, Subject, Examination, Remark, Result, Class, Section, Attendance and Attendance record.   
Here, one school consist of many students so school is represented as one to many relationship student. One class can have many sections, so it is thereby shown as one to many relationship. Under one examination there are several subjects so this is one to many relationship.

We needed to make attendance base upon the subjects i.e. subject wise attendance is to be performed. And based upon this issue, on the one attendance record there are multiple subjects. This mean subject have many to one relationship with attendance record entity.

Result table consist of all the students name, class, section, examination type, marks obtained on different subjects and so on. One result can have multiple subjects (one to many), one examination can have multiple results (one to many) and multiple remarks can be in one result (one to many).

Establishing all the possible relationship with entities can be fruitful to make database of the system.

## **Architectural Model**

**Client Application**

Database User

Database Application

**Server**

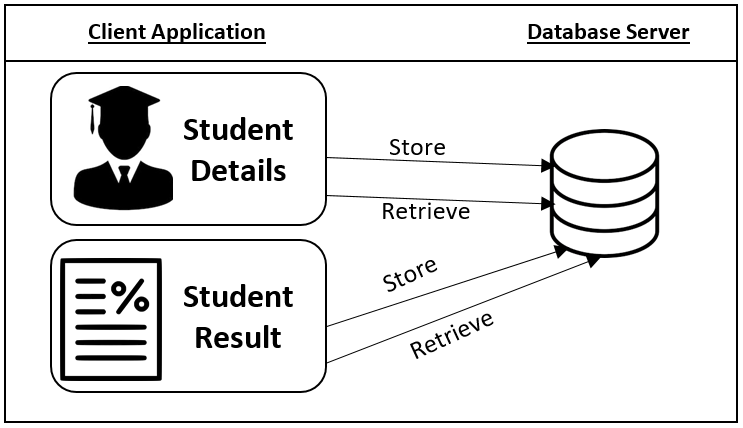
Database System

The two- tier architecture is simple type of architecture where the system is actually divided into two layers. One layer is called Client layer and another is Database layer.

Client layer consist of client or user application i.e. application that user will be using.

Database layer mean database of the system where data is being store.

In this architecture, Client action is taken in action after firing database query into the database through application connection interface like JDBC, ODBC, etc.

  
For instance, in this system Mark Sheet Generator, when user enters the student information which is to be stored in database, it is stored in database by insert query at the back end of the program. Similarly, when the information is to be retrieve, select query is being trigger. In this way, other functionality of this system works on using two- tier architecture.

## **UI Modelling**

## **Prototype:**

Prototype is sample model of the system that is use to make client understand about their system and developer to know about the requirements.

## 

## 

