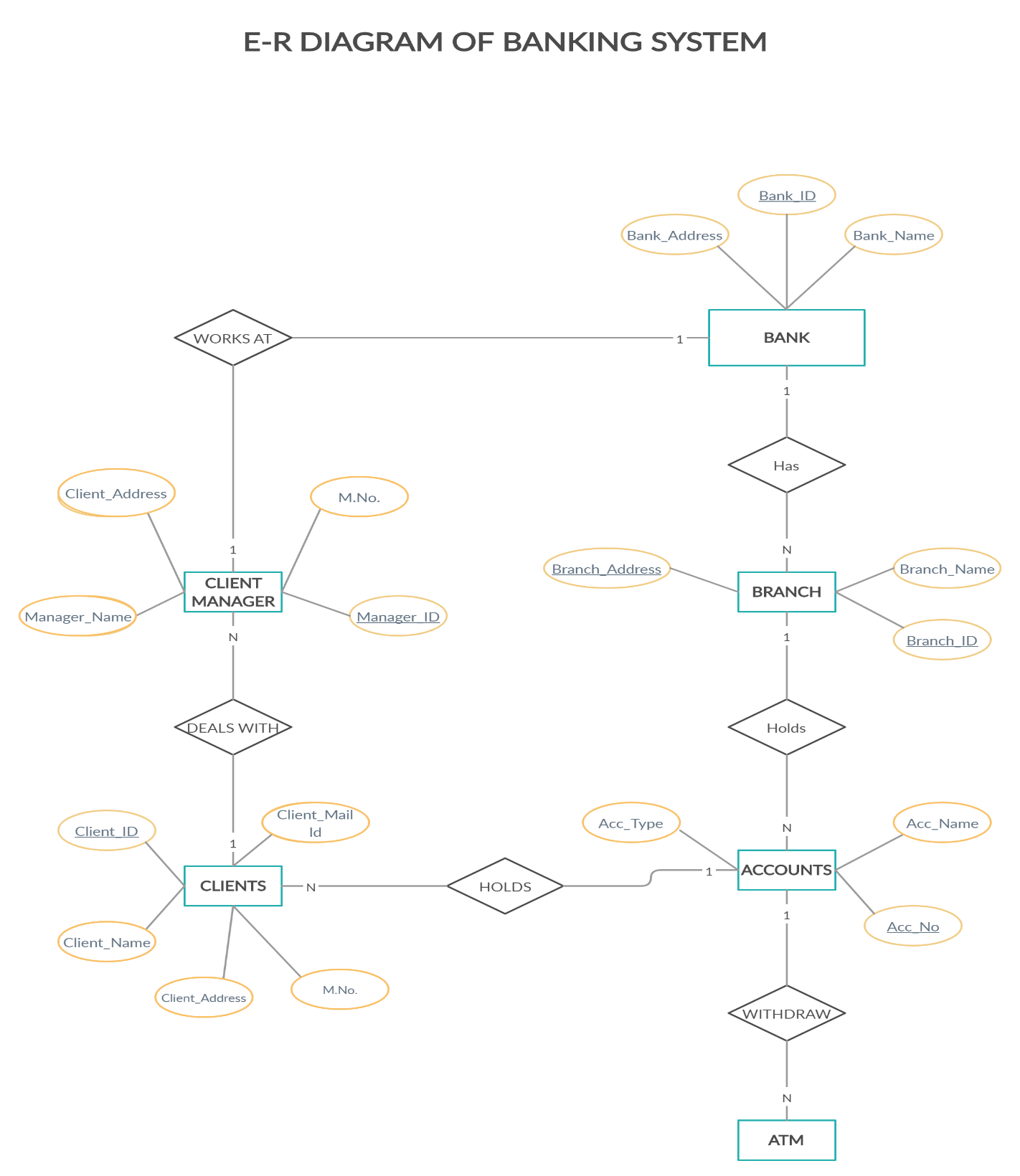
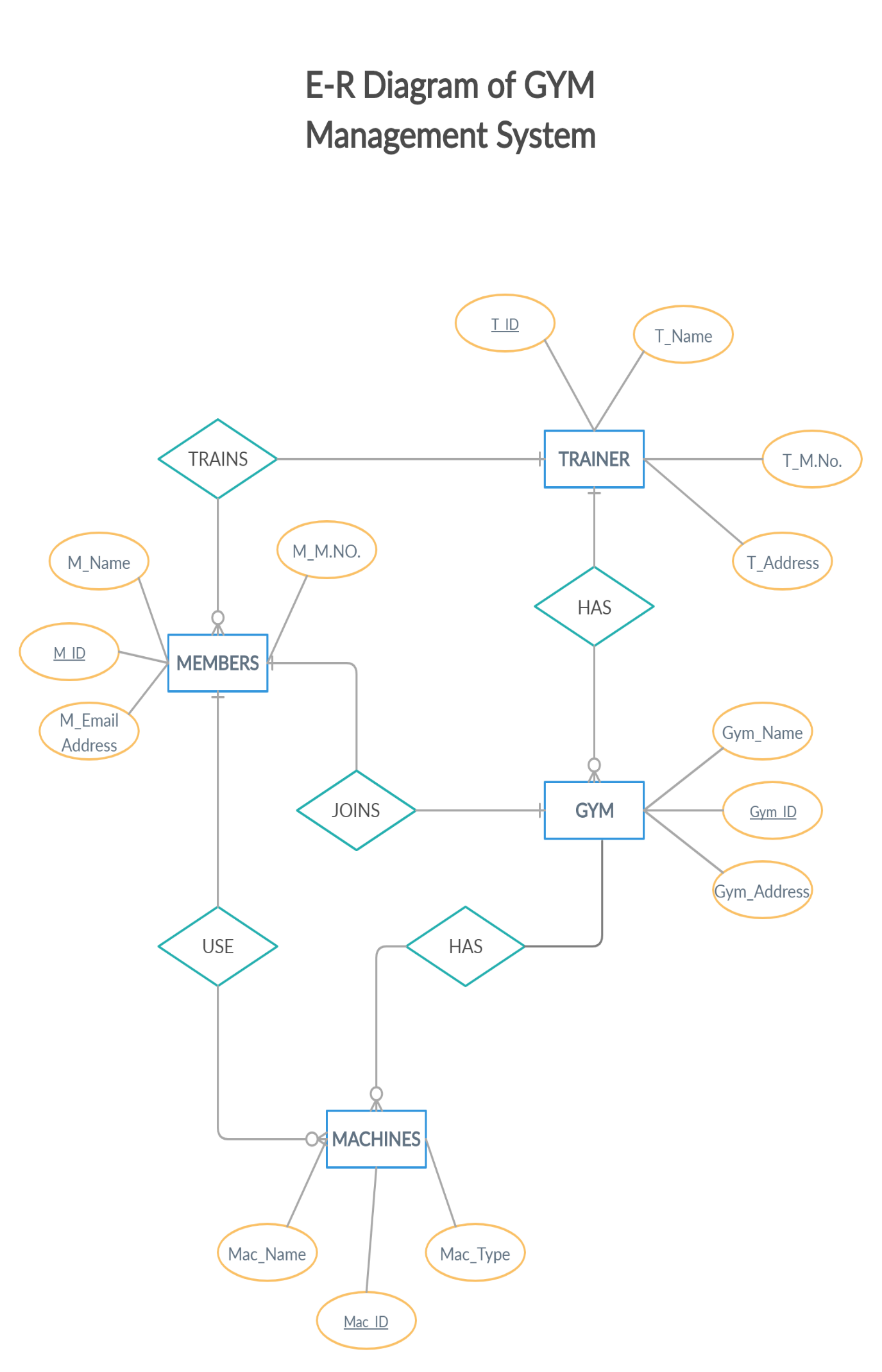
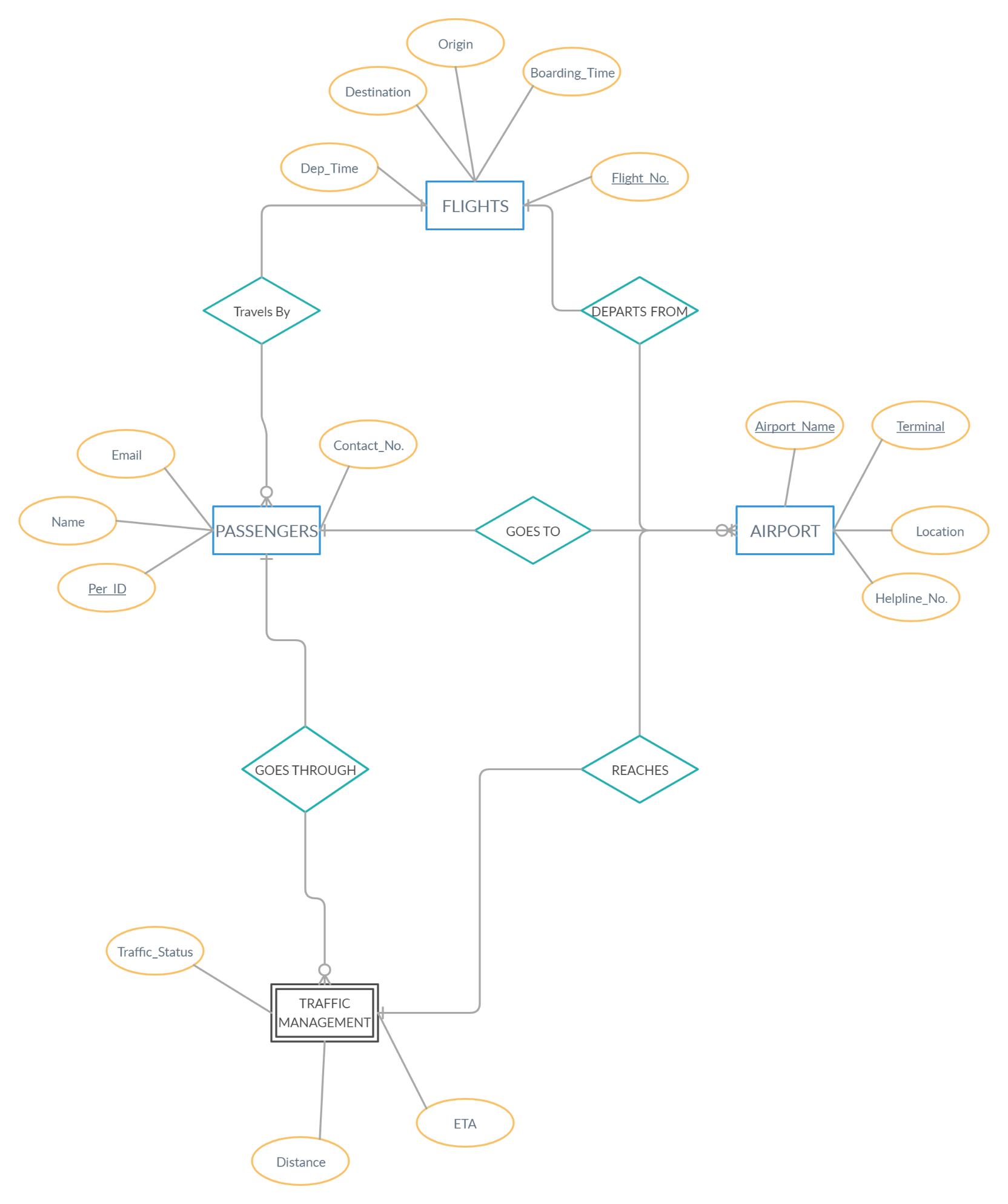
1. **PROBLEM STATEMENT:**

The airport management system is basically an all-in-one mobile application for your complete flight experience, from home to the airport. It enables the users to get details regarding the timings of their flight, the boarding gate, traffic conditions from their residence to the airport, to call for assistance and also regarding the in-airport amenities such as restaurants, shopping centres etc.

1. **ER DIAGRAMS:**
2. ****
3. ****



1. **ER DIAGRAM- AIRPORT MANAGEMENT SYSTEM**

**SOFTWARE REQUIREMENT SPECIFICATIONS**

**AIRPORT MANAGEMENT SYSTEM**

**1.INTRODUCTION**

**1.1 PURPOSE**

The purpose of this document is to describe the Airport Management System. It contains the functional, behavioural and non- functional requirements of the project and contains guidelines for system engineers and designers to start working on this project.

**1.2 SCOPE**

The airport management system is basically an all-in-one mobile application for your complete flight experience, from home to the airport. It enables the users to get details regarding the timings of their flight, the boarding gate, traffic conditions from their residence to the airport, to call for assistance and also regarding the in-airport amenities such as restaurants, shopping centres etc.

**2. PRODUCT FUNCTIONS**

**2.1 ADMINISTRATOR**

The admin should be able to –

* Modify information about the flight timings ie. Boarding time, departure time, flight delays if any, approximate waiting time for baggage and security check etc.
* Modify details about flight such as flight number
* Contact the users in case of flight cancellations or other issues
* Get information about users who have upcoming flights
* Add and edit information about airport amenities
* Send reminders or late warning to users with upcoming flights
* Respond to any assistance calls by the user and further contact higher authorities if needed

**2.2 NORMAL USERS (PASSENGERS)**

The passengers should be able to –

* Get all required details about their flights
* Get information about traffic conditions from their residence to the airport
* Change their personal details such as contact number, name, email address etc.
* Should be able to contact airport authorities for assistance
* Get information about restaurants, shopping centres etc at the airport
* Give feedback or complaints regarding the application to the admin

**2.3 OPERATING ENVIRONMENT**

This is a mobile based application and should be available on all major platforms such as IOS, Android, Windows, Blackberry etc.

**2.4 USER CHARACTERISTICS**

The administrators are airport authority personnel, who are expected to have knowledge about the internal modules of the system and should be able to rectify small problems that may arise due to disk crashes, power failures or other catastrophes. The normal users are passengers who have upcoming flights, and are assumed to have basic knowledge about mobile phones and applications.

**2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS**

* Information must be stored in a database accessible by the airport authorities
* MySQL will be used as SQL engine and database
* The system should be active 24\*7
* The users may access their accounts from any mobile phone, provided it has the mobile application and an active internet connection
* Users must be allowed to log into their accounts provided they have the correct username and passwords

**3. EXTERNAL INTERFACE REQUIREMENTS**

**3.1 USER INTERFACES**

* LOGIN INTERFACE : The user should be able to login using his username and details, if already registered. In case of incorrect username or password, an error message should be displayed. Unregistered users should be able to register by entering their details.
* CATEGORIES : The user should be able to the browse through the different categories in the app, such as the flight details section, Personal Info section, Call for assistance section, Traffic details section and the amenities section
* SEARCH INTERFACE: The user should be able to search and browse through the availability of food and shopping centres at the airport
* ADMIN CONTROL PANEL : The admin should be able to check details of passengers and update any required information about the flight.

**3.2 HARDWARE INTERFACES**

No extra hardware required except a smartphone with active an internet connection.

**3.3 SOFTWARE INTERFACES**

Only the mobile operating system is required

**4. FUNCTIONAL REQUIREMENTS**

**4.1 PASSENGERS**

* Register to the application
* Get traffic details
* Call for assistance

**4.2 ADMIN**

* Remove users if required
* Add or edit information about flights
* Add or edit details about airports
* Contact users if required

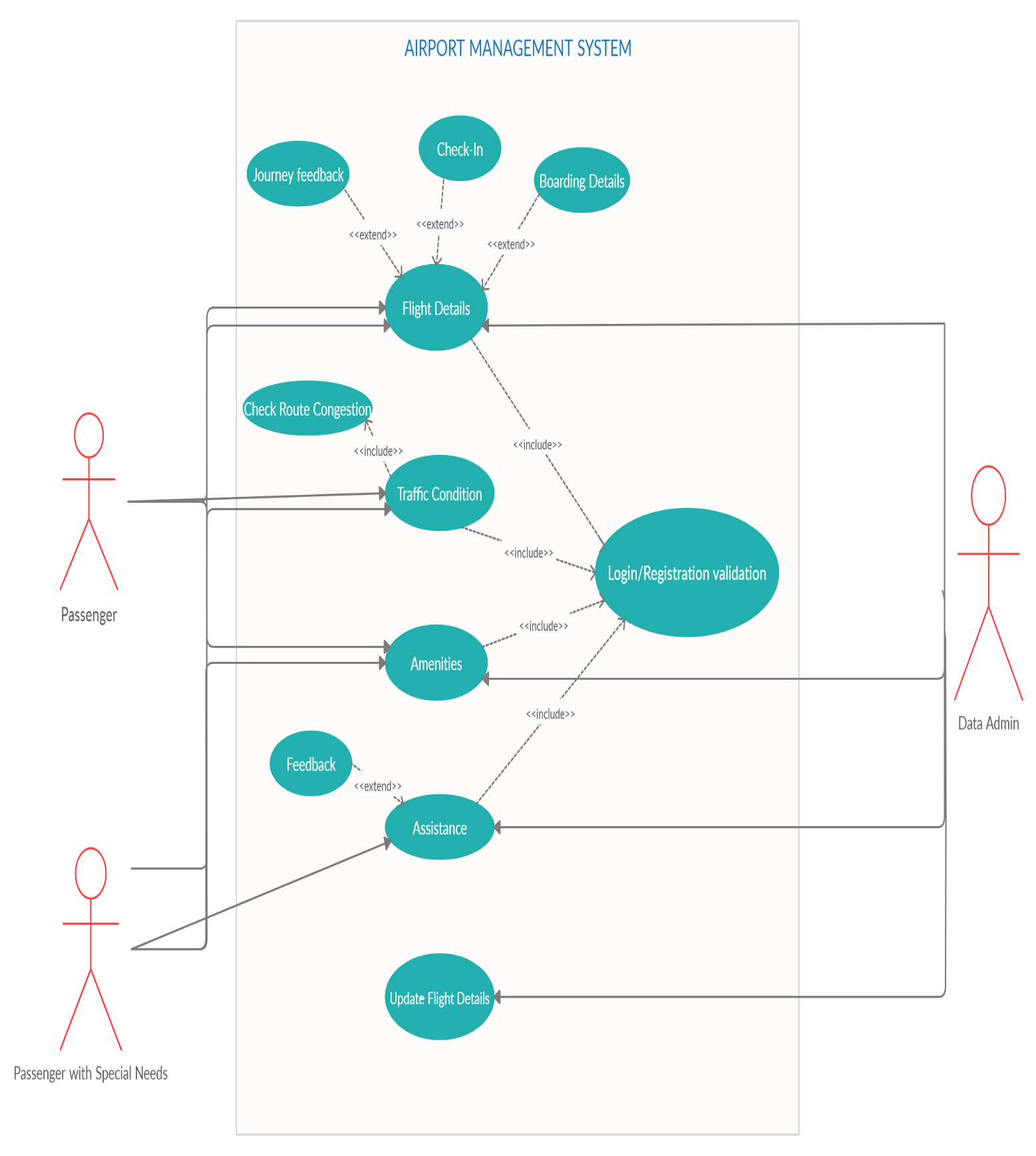
**4.3 COMMON FUNCTIONS**

* Login
* Search for flight details

**5. NON-FUNCTIONAL REQUIREMENTS**

* ERROR HANDLING : The system should be able to handle expected and unexpected errors in ways that prevent loss in information and long downtime periods
* PERFORMANCE REQUIREMENTS : The system should be able to handle a large number of users without any negative effects on the speed and functionality of the system.
* SAFETY REQUIREMENT : The application should not be the cause of harm to the users
* SECURITY REQUIREMENTS : The system should use a secured database. The passengers should be able to view information but not edit it, except for their personal info. The system will have different types of users and both users shall have a specific set of constraints.

**4) USE CASE DIAGRAM:**

****

**5)**

**SOFTWARE DESIGN DOCUMENT**

**AIRPORT MANAGEMENT SYSTEM**

**1.INTRODUCTION**

**1.1 PURPOSE**

The purpose of this document is to describe the Airport Management System. It contains the functional, behavioural and non- functional requirements of the project and contains guidelines for system engineers and designers to start working on this project.

**1.2 SCOPE**

The airport management system is basically an all-in-one mobile application for your complete flight experience, from home to the airport. It enables the users to get details regarding the timings of their flight, the boarding gate, traffic conditions from their residence to the airport, to call for assistance and also regarding the in-airport amenities such as restaurants, shopping centres etc.

**1.3 OVERVIEW**

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. It provides a high level system architecture, showing a subsystem breakout of the system. It shows the interfaces to the external systems.

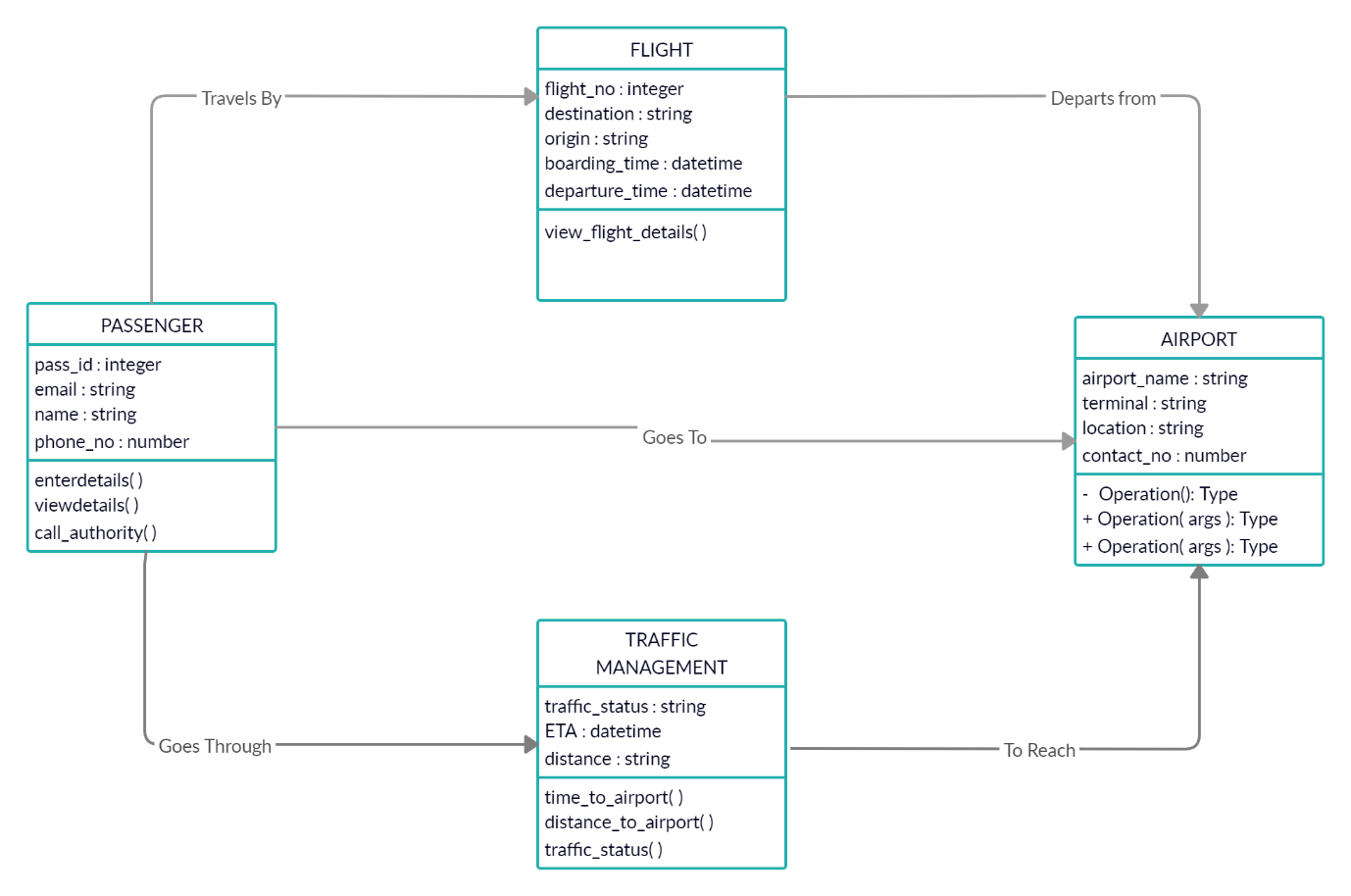
**2. SYSTEM OVERVIEW**

This system functions on 2 levels :

1. For passengers : It enables the passengers to access all the required information about their flight in one go. The passengers can login into their accounts using any mobile device, provided it has an active internet connection and has this application installed.
2. For Administrators : The administrators can input and edit information about the flights, the airport and the amenities available at the airport. A 2-step authorisation is required to log into the administrator’s account.

**3. SYSTEM ARCHITECTURE**

**3.1 ARCHITECTURAL DESIGN**



**3.2 DECOMPOSITION DESCRIPTION**

The system is decomposed into 4 modules, which have the following functions:

1. PASSENGER : This module represents the passengers travelling by the flights. The associated functions are:

* Enterdetails(): this function is used to get the details of a passenger ie. Name, email, contact number etc.
* Viewdetails(): This function is used to display all the details of a given passenger.
* Call\_authority(): This function is used to contact the authorities of a particular airport in case of emergencies.

1. FLIGHT : This module represents the flights departing from the airport. The associated functions are:

* Enter\_flight\_details(): this function is used to get the details of the flight ie. Flight number, destination, origin, boarding time etc.
* View\_flight\_details(): This function is used to display all the details of a given flight.

1. AIRPORT : This module represents the Airports. The associated functions are:

* Enter\_airport\_details(): this function is used to get the details of a particular airport ie. Name, terminal, contact number etc.
* Viewdetails(): This function is used to display all the details of a given airport.

1. TRAFFIC MANAGEMENT : This module reflects the traffic conditions enroute the residence of a passenger to the airport. The associated functions are:

* Time\_to\_airport(): this function is used to get the time required to get to the airport.
* Distance\_to\_airport(): This function is used to display the distance to the airport.
* Traffic\_status(): This function is used to get the status of traffic at any given time from the airport to the residence.

**3.3 DESIGN RATIONALE**

This decomposition was preferred over the others as it provided much better separation of various modules while still keeping them properly functional. The required data could be easily accessed without any redundancy and it helped in giving an easy overview of the project as it separates the major components of the system.

**4. DATA DESIGN**

**4.1 DATA DESCRIPTION**

MySQL is used as the structured query language to store and retrieve information from the database in this project. The database is divided into 4 tables, one for each of the decomposed modules ie. Passenger, Airport, Flight and Traffic Details.

**4.2 DATA DICTIONARY**

1. PASSENGER

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **CONSTRAINTS** |
| pass\_id | Integer | Primary key |
| name | String | Not Null |
| email | String | Not Null |
| phone\_no | Number | Unique |

1. FLIGHT

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **CONSTRAINTS** |
| flight\_no | Integer | Primary key |
| destination | String | Not Null |
| origin | String | Not Null |
| boarding\_time | Datetime | Not Null |
| departure\_time | Datetime | Not Null |

1. AIRPORT

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **CONSTRAINTS** |
| name | String | Primary key(part 1) |
| terminal | String | Primary key(part 2) |
| location | String | Not Null |
| contact\_airport | Number | Unique |

4)TRAFFIC MANAGEMENT

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **CONSTRAINTS** |
| traffic\_status | String | Not Null |
| distance\_to\_airport | String | Not Null |
| ETA | Datetime | Not Null |

**5. COMPONENT DESIGN**

1. PASSENGER : This module represents the passengers travelling by the flights. The details of associated functions are:

* Enterdetails(): this function is used to input the details of a passenger into the database using mysql queries.

Example-

insert into passenger Values(2243, ‘Karan’, ‘karanmalik@abc.com’, 9898899658);

* Viewdetails(): This function is used to extract the details of a passenger from the database.

Example-

Select \* from passenger

Where pass\_id=2243;

* Call\_authority(): This function is used to contact the authorities of a particular airport in case of emergencies.

1. FLIGHT : This module represents the flights departing from the airport. The details of associated functions are:

Enter\_flight\_details(): this function is used to input the details of a flight into the database using mysql queries.

Example-

insert into flight Values(22465, ‘Paris’, ‘New Delhi’, 22:00, 23:15);

Viewdetails(): This function is used to extract the details of a passenger from the database.

Example-

Select \* from flight

Where flight\_no=22465;

1. AIRPORT : This module represents the Airports. The details of the associated functions are:

Enterdetails(): this function is used to input the details of an airport into the database using mysql queries.

Example-

insert into airport Values(‘IGI’, ‘Terminal-3’, ‘New Delhi’, 45454545);

Viewdetails(): This function is used to extract the details of an airport from the database.

Example-

Select \* from airport

Where airport\_name=’IGI’ AND terminal=’Terminal-3’;

1. TRAFFIC MANAGEMENT : This module reflects the traffic conditions enroute the residence of a passenger to the airport. The associated functions are:

* Time\_to\_airport(): this function is used to get the time required to get to the airport from the database.

Example-

Select ETA from TRAFFIC\_MANAGEMENT ;

* Distance\_to\_airport(): This function is used to display the distance to the airport.

Example-

Select distance\_to\_airport from airport ;

* Traffic\_status(): This function is used to get the status of traffic at any given time from the airport to the residence.

Example-

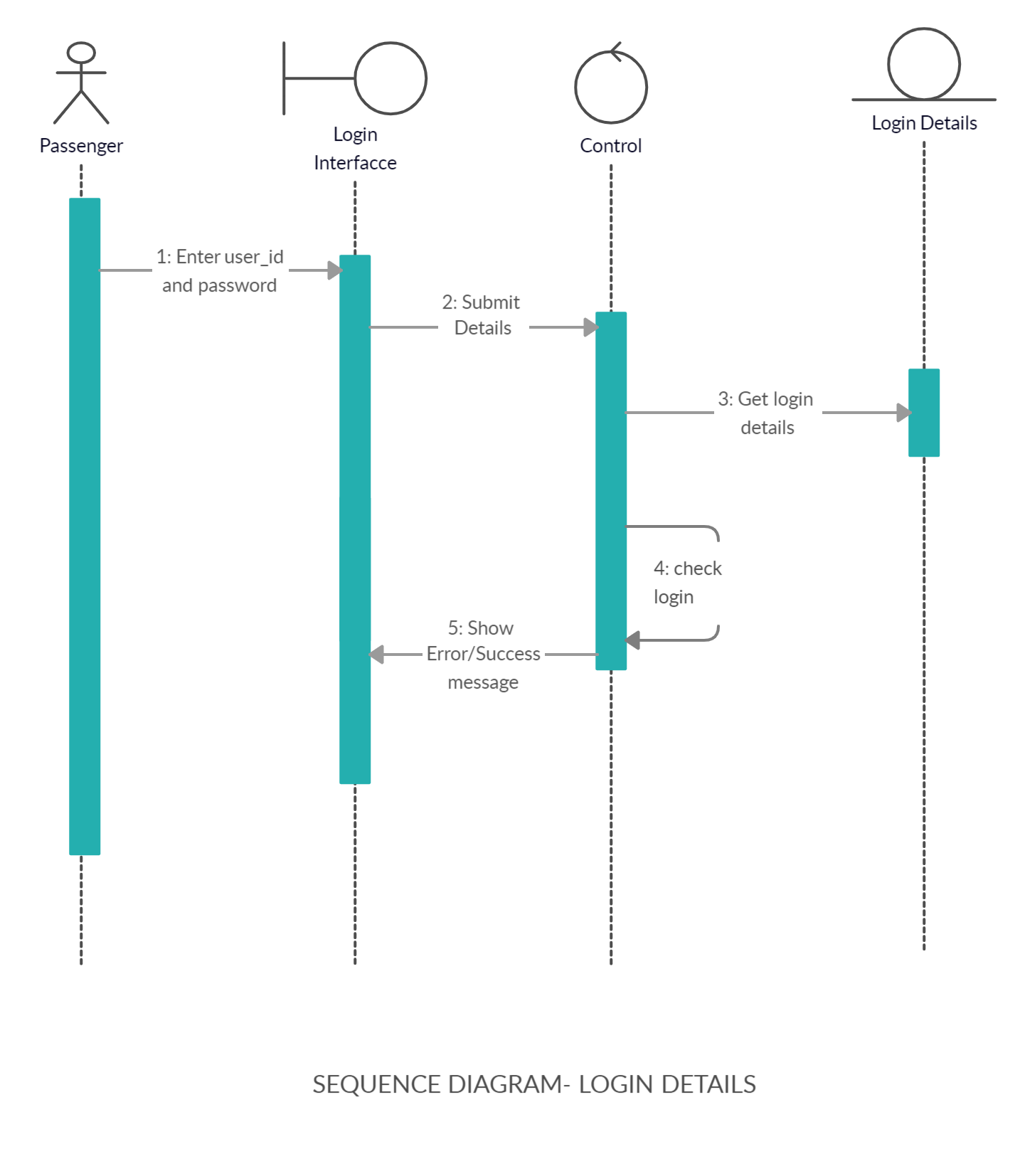
Select traffic\_status from airport ;

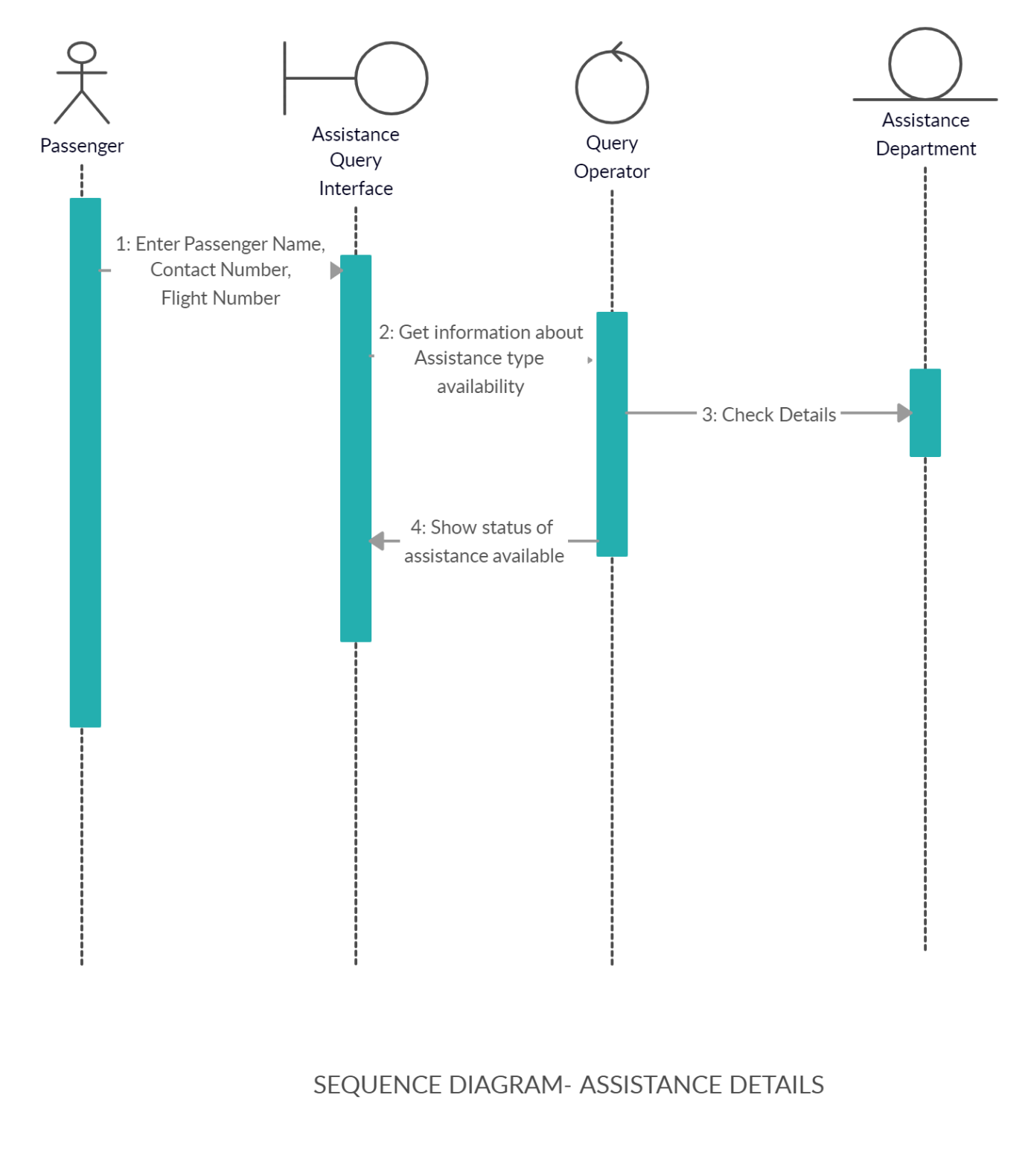
**6.USER INTERFACE**

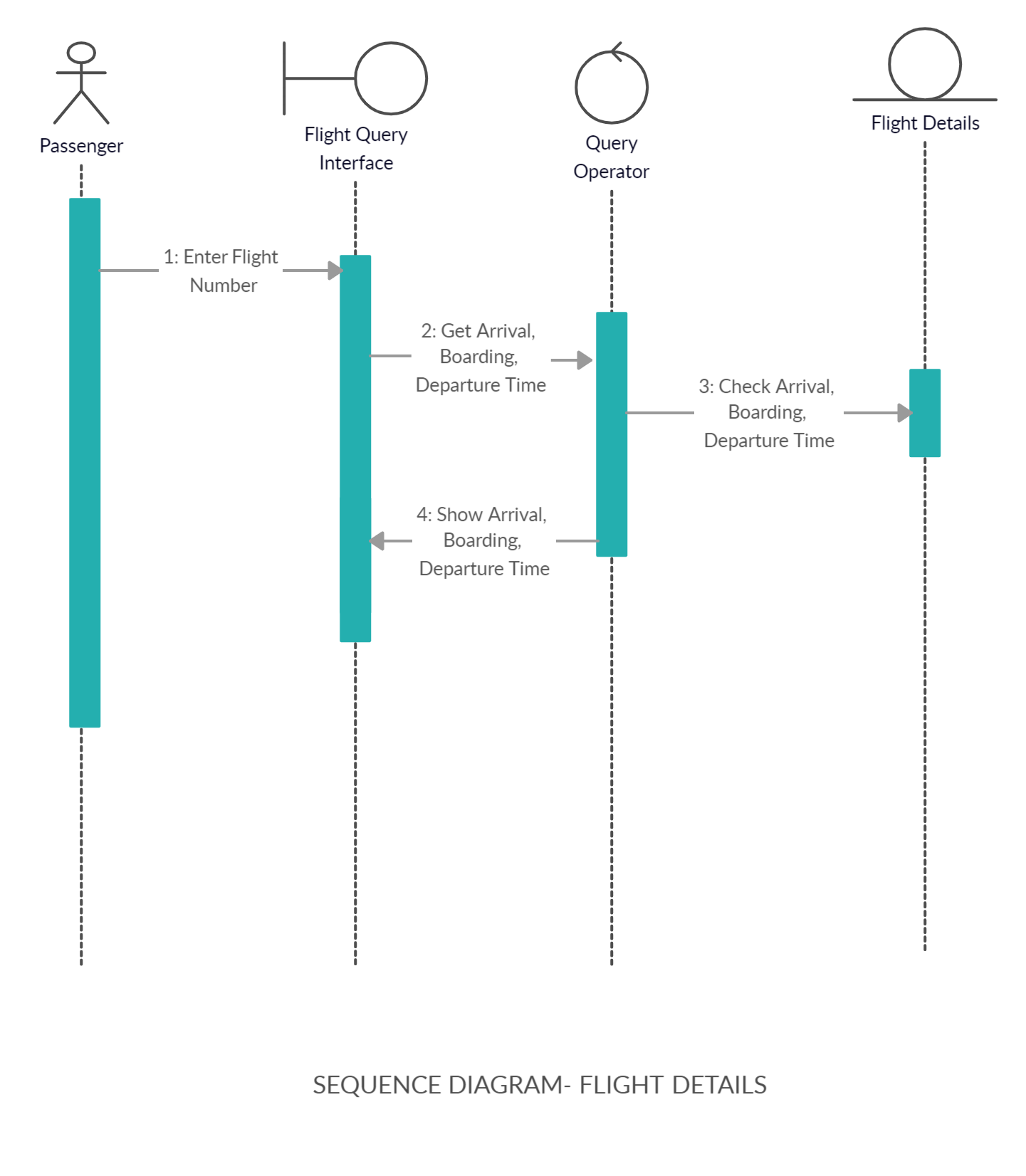
The user will be able to login into his/her account using the login option on the home page of the application. Without logging in the user will still be able to access a few selected functionalities of the application, such as viewing flight details and airport amenities. If the correct username and password is provided, the user gets logged into his account else an error message is displayed. After logging in the user can enter and edit his details, along with the previous mentioned functionalities.

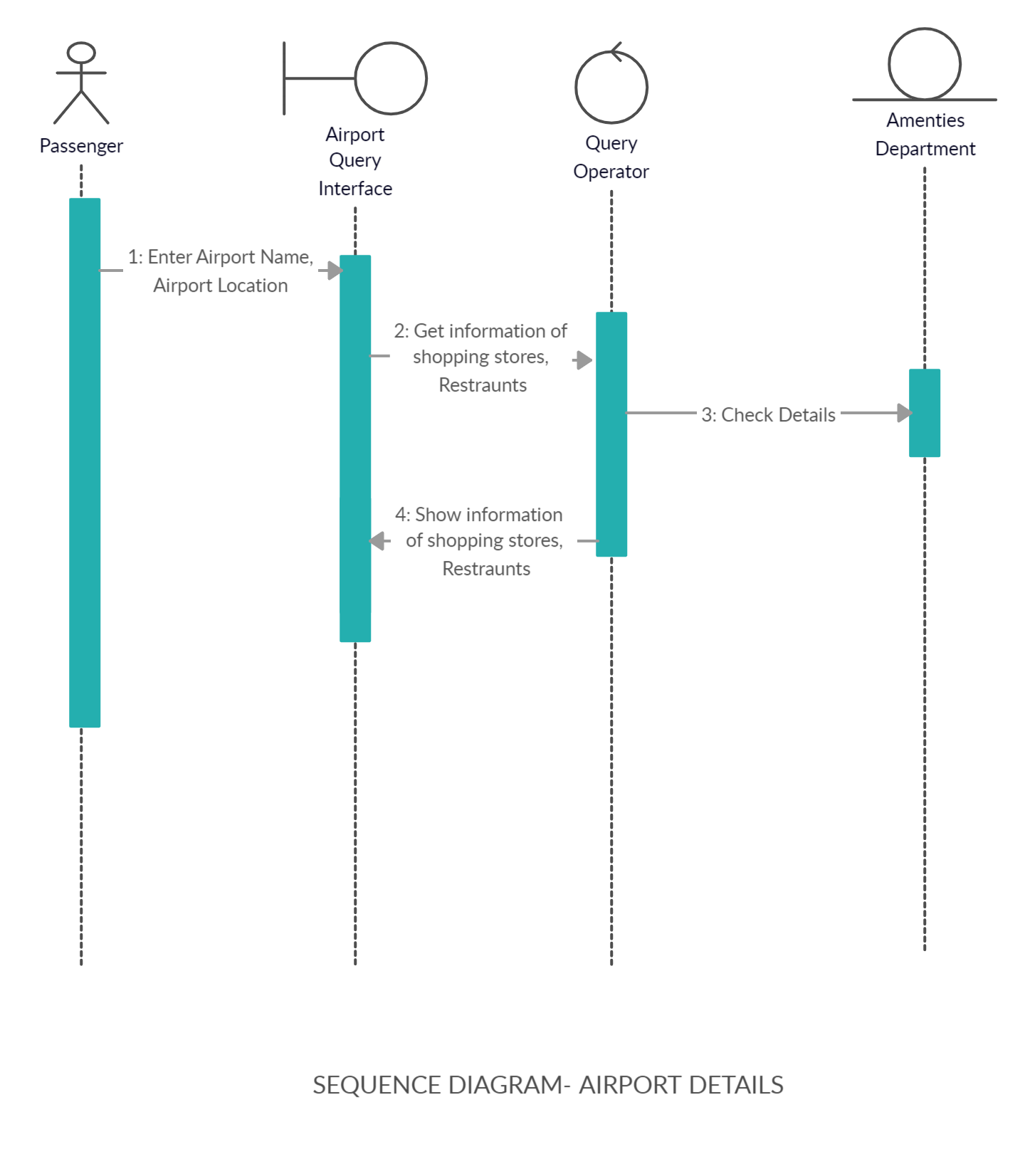
The user can provide feedback using the provide feedback option in the settings, provided he is logged in.

1. SEQUENCE DIAGRAMS:

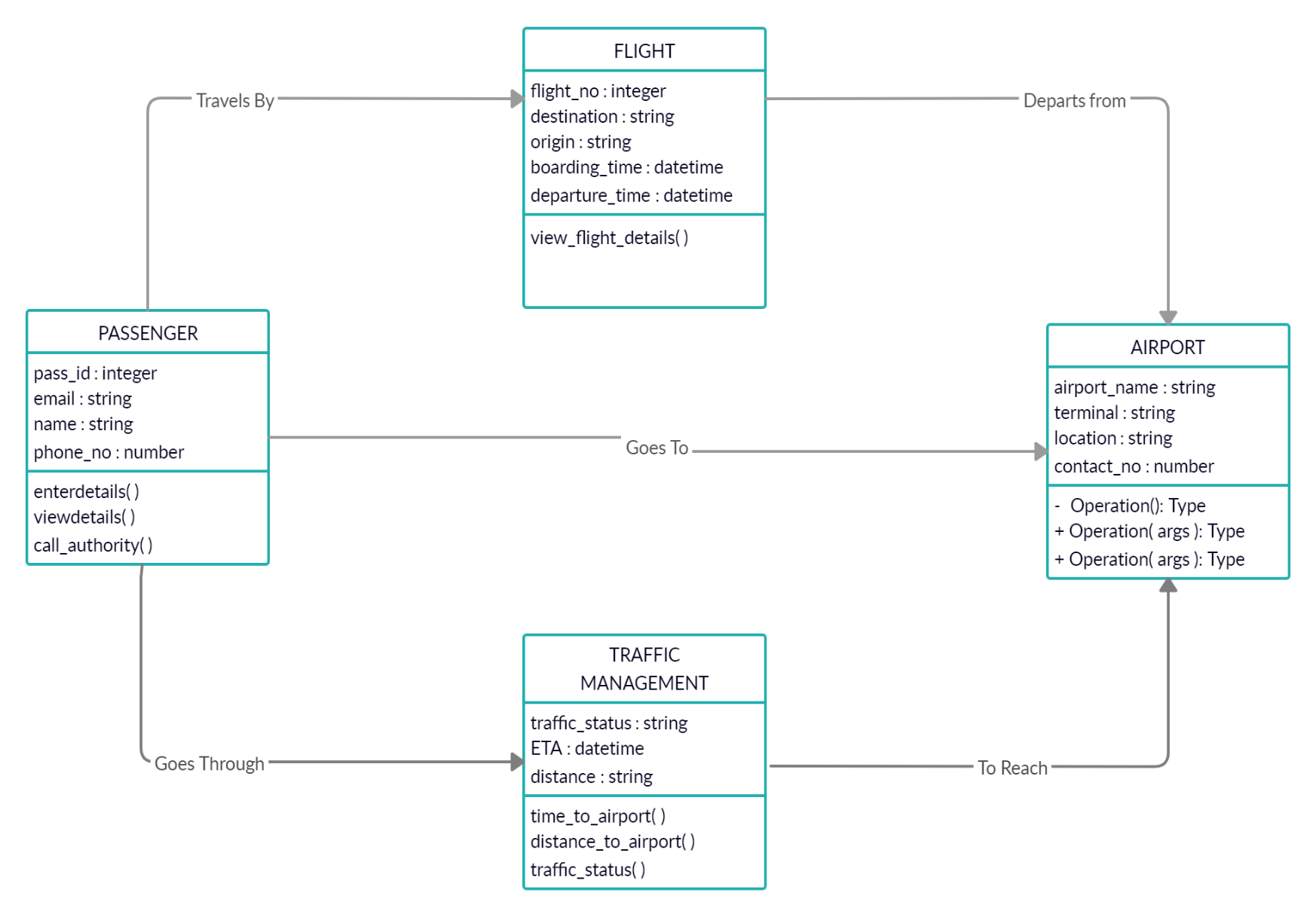








1. CLASS DIAGRAM:



1. DFD:

