

BANASTHALI VIDYAPITH



Apaji Institute Of Mathematics & Applied Computer Technology

Software Design Specifications

For

College Enquiry System with Chatbot - AIDA

Group ID: BTI_G19

(Dated: 25th Jan 2022)

Guided By:

Dr Karuna Sharma

Submitted By:

1. Aditi Jaiswal(2016699) IT
2. Avni Jindal(2016728)IT
3. Deeksha Aggarwal(2016734)IT
4. Ismit Singh(2016752)IT

Table of Contents

1. Introduction	1
1.1 Purpose of this document	1
1.2 Scope of the development project	1
1.3 Definitions, acronyms, and abbreviations	1-2
1.4 References	2
1.5 Overview of document	2
2. System Architecture Description	2
2.1 Overview of modules/components	2
2.2 Structure and relationships	3-10
3. Detailed Description of Components	11
3.1 Data Dictionary	11
4.0 Design Decisions And Trade-Offs	12
5.0 Pseudo-Code For Components	12-13

1. Introduction

1.1 Purpose of this document

The purpose of SDS document includes an architectural design of “**AIDA**”. It includes data design, detail design and interface design. It provides a description of the design of the website fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to built.

1.2 Scope of the development project

“**AIDA**” is a chatbot based website which provides users with answers to all of their queries related to Banasthali Vidyapith.

The scope of the project can be listed as under:

- Administrators will be responsible to maintain the overall database of the website.
- Users will be able to ask their queries to the chatbot.
- Students will be able to ask their classes, etc related queries to chatbot.
- Chatbot will match the queries and answer accordingly. If any query doesn't match, it will show the “sorry message”.

Note: Users of the website can be **Student, Parents, Admission seeker** and **Administrator**. All the users will have different access and rights granted by this website.

1.3 Definitions, acronyms, and abbreviations

1.3.1 Definitions

- **Class diagram-** It is a static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationship among objects.
- **Activity Diagram-** Activity Diagram illustrate the dynamic nature of a system by modelling the flow of control from activity to activity.
- **Sequence Diagram-** UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.
- **Use Case Diagram-** A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

- **Data-Flow diagram-** It is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

1.3.2 Acronyms and Abbreviations

- **SDS:** Software Design Specifications
- **SRS:** Software Requirements Specifications

1.4 References

Books:

- Software Engineering by KK Agarwal
- Software Engineering by KK Agarwal & Yogesh Singh

1.5 Overview of document

- Section 2 provides a general description of the functionality, context ,design of the system and architectural design explaining the subsystems with the help of various UML diagrams.
- Section 3 provides the detailed description of components.
- Section 4 provides the overview of database design along with the description of system entities.
- Section 5 provides the design of the user interface.

2. System architecture description

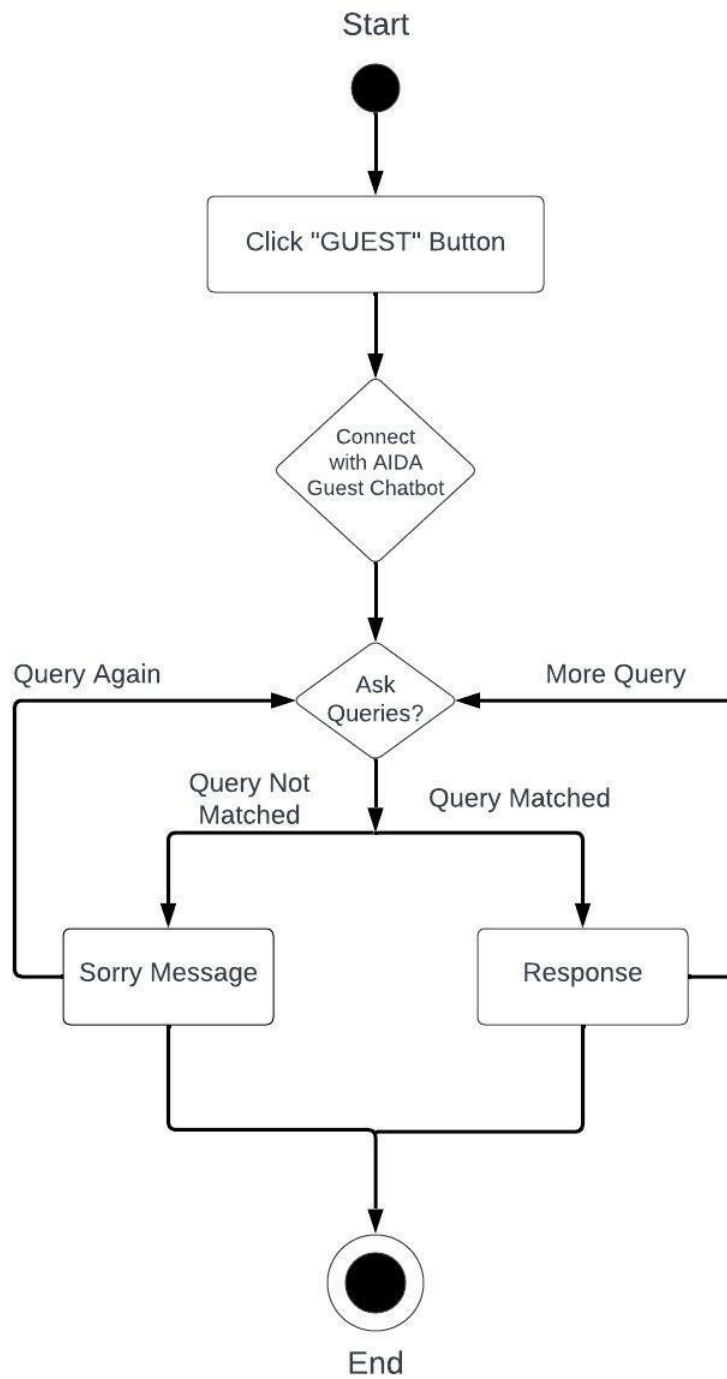
2.1 Overview of modules / components

- **User:** Any guest user can directly access to chatbot by single-button click.
- **Student Login:** Students need to login with their Banasthali IDs and passwords in order to access Student Chatbot..
- **AIDA Guest Interaction:** Guest can ask query. If query will match with chatbot database, response will be received otherwise sorry message will be received.
- **AIDA Student Interaction:** Student can ask query. If query will match with chatbot database, response will be received otherwise sorry message will be received.

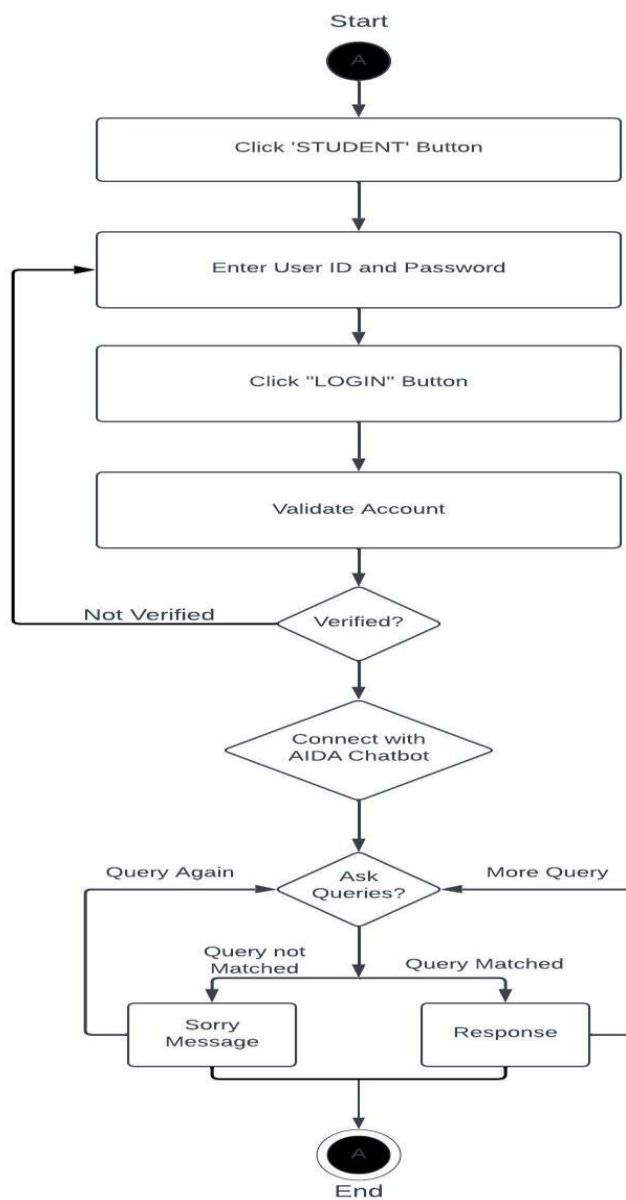
2.2 Structure and relationships

ACTIVITY DIAGRAMS:

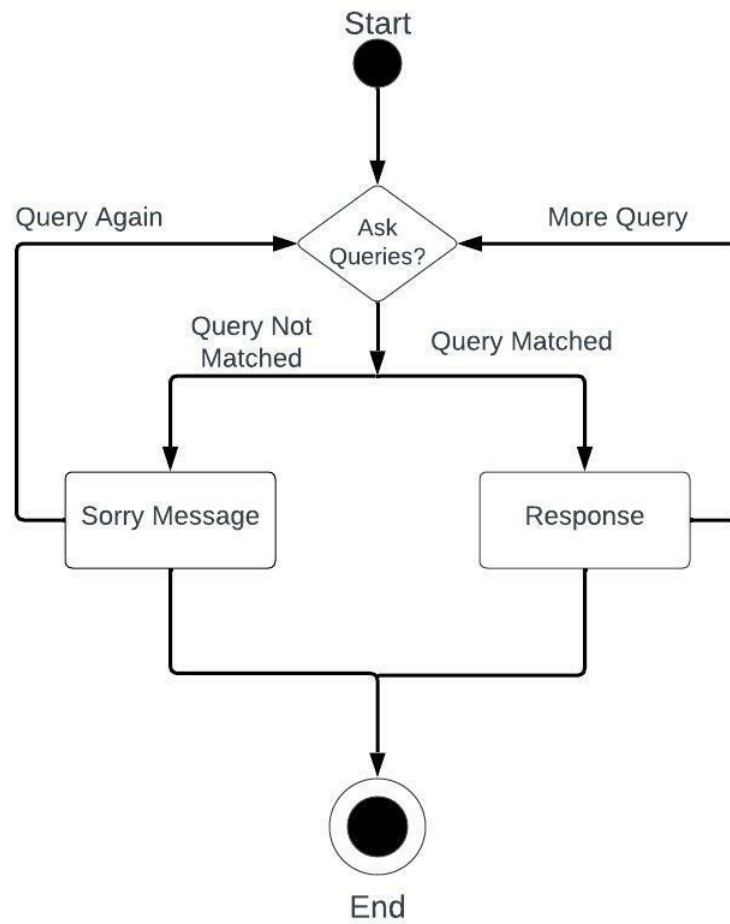
- **GUEST Access**



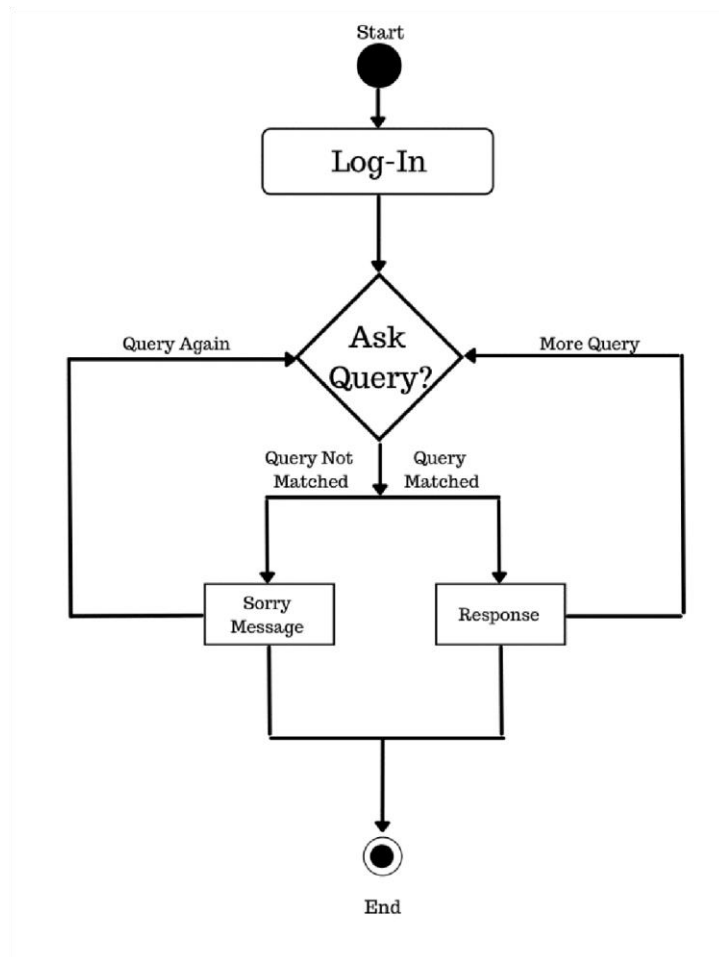
- **STUDENT Login and Interaction**



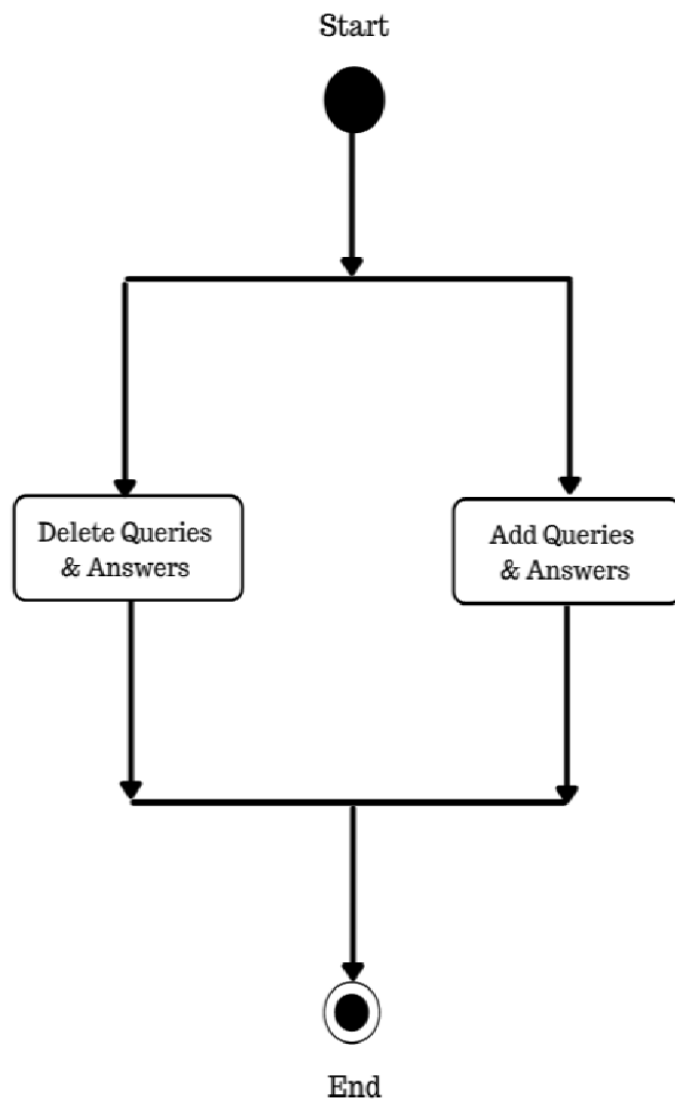
- GUEST User functionalities**



- **Authorized (Student) User functionalities**

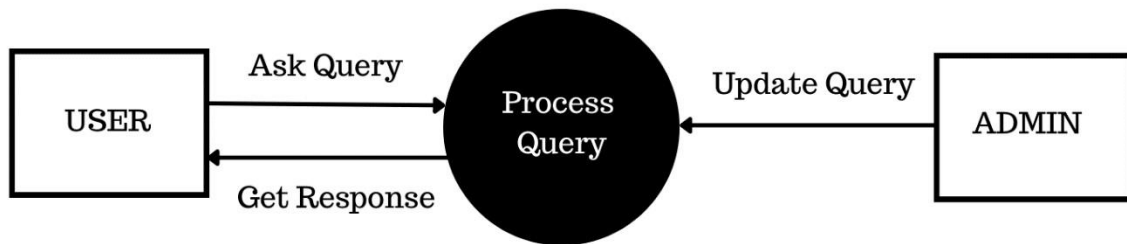


- **Admin functionalities :**

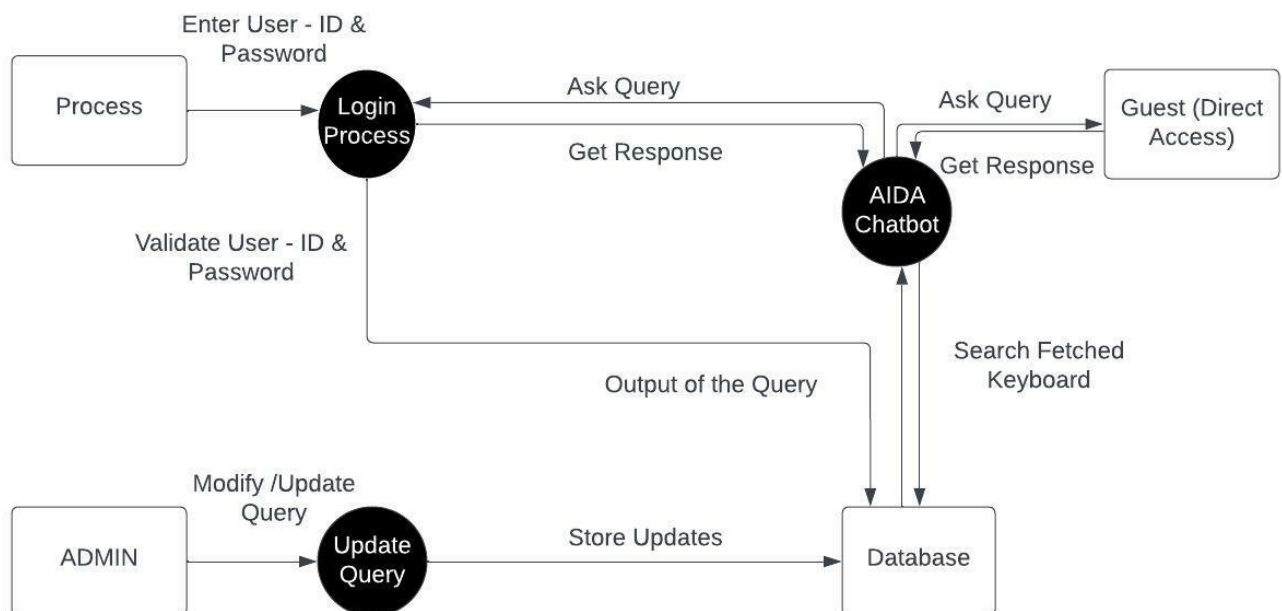


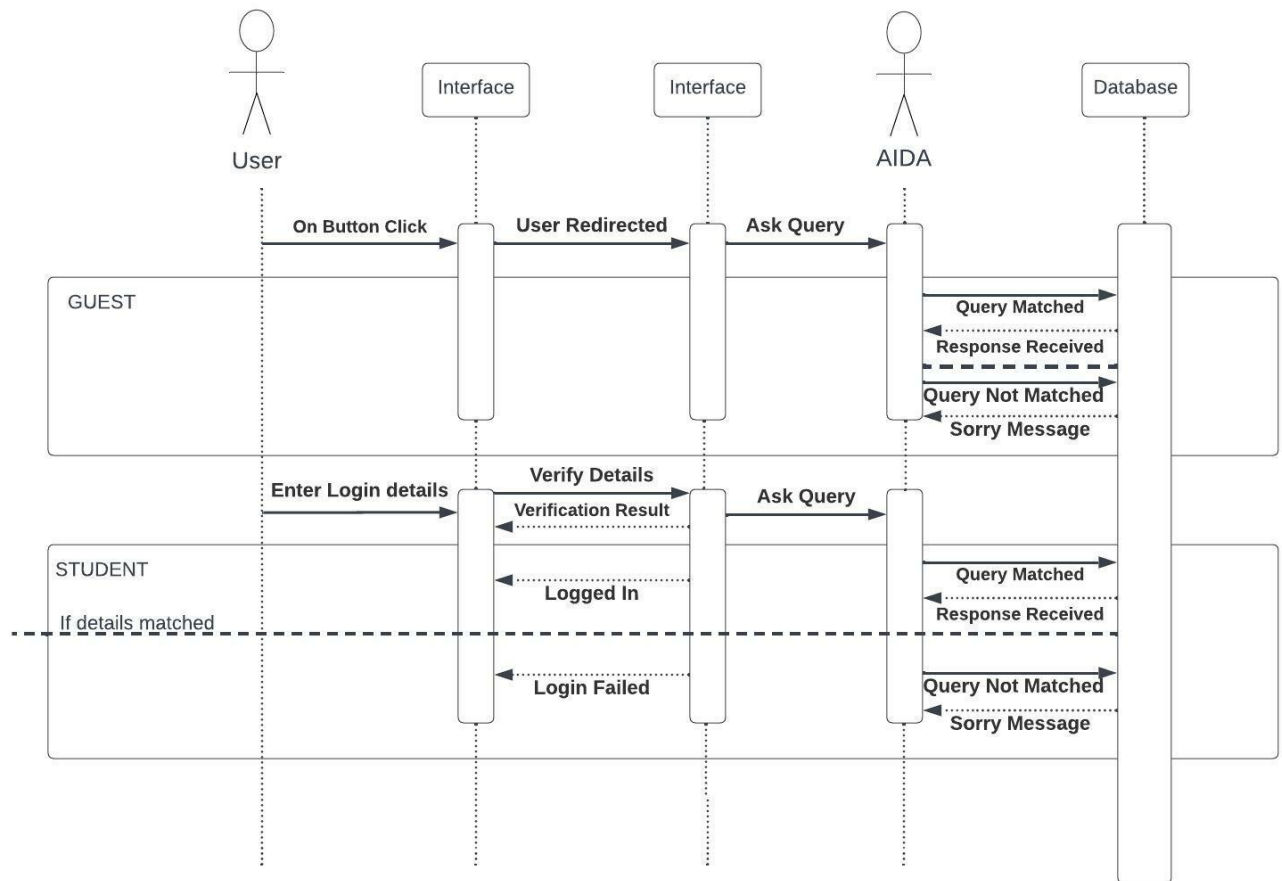
DATA FLOW DIAGRAMS:

• 0 Level DFD:

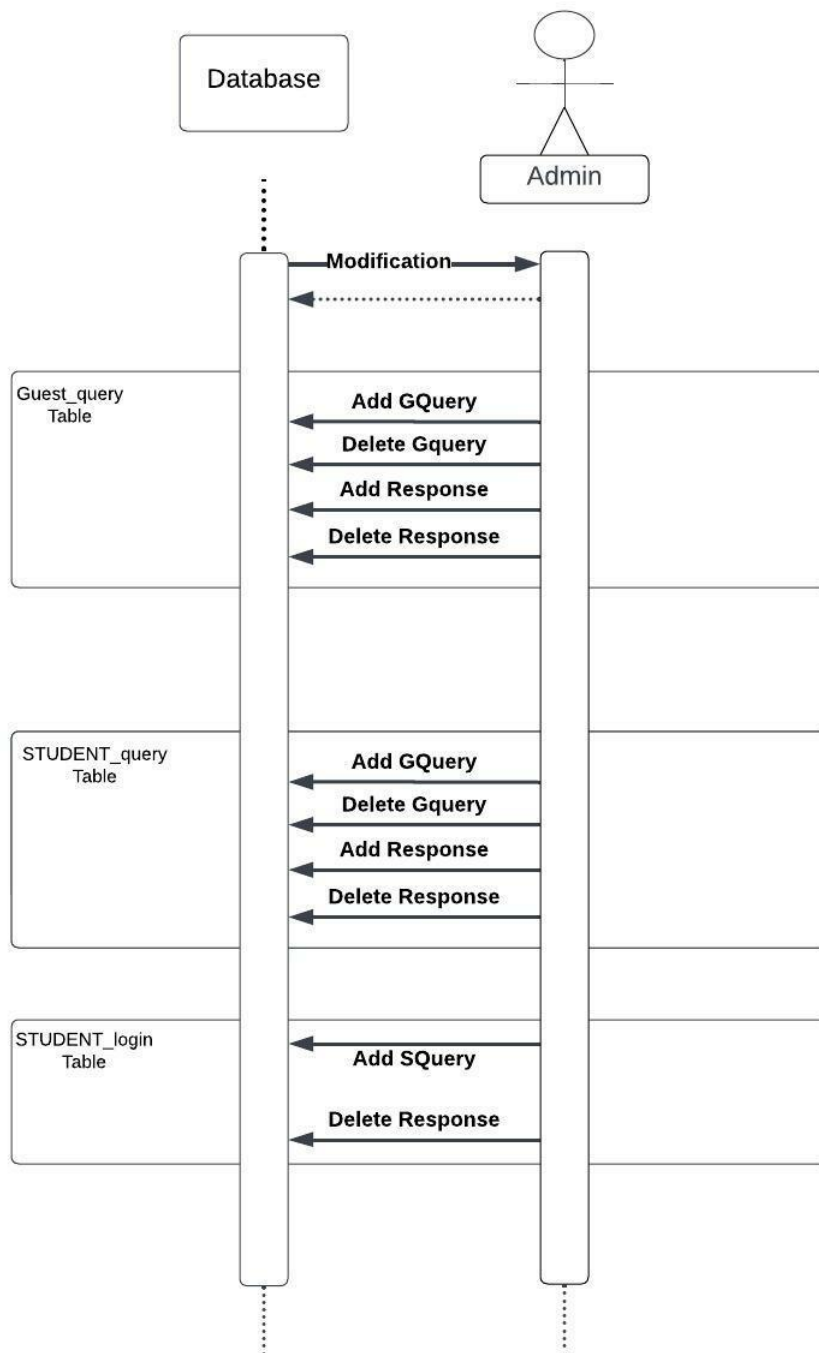


1 LEVEL DFD:



SEQUENCE DIAGRAM FOR LOGIN AND QUERY:

SEQUENCE DIAGRAM FOR ADMIN:



3. Detailed description of components

3.1 DATA DICTIONARY

- **Student_details**

Field Name	Data Type	Size	Constraints
Student-Id	VARCHAR	10	Primary Key
S_Password	VARCHAR	10	NOT NULL

- **GUEST_query**

Field Name	Data Type	Size	Constraints
G_Query	VARCHAR	1000	Primary Key
G_Response	VARCHAR	1000	NOT NULL

- **STUDENT_query**

Field Name	Data Type	Size	Constraints
S_Query	VARCHAR	1000	Primary Key
S_Response	VARCHAR	1000	NOT NULL

4.0 Design decisions and trade-offs

- FAQs can also be added by Users (Guest, Student, Faculty). But this idea is abandoned, because queries and responses are need to be verified by Admin before adding to the database. So that only correct information will be provided to the users.
- Website Design depends on the devices.

5.0 Pseudo-code for components

- **LOGIN (Student):**

```
if ($userid==$entered_userid && $password==$entered_password)
echo "Login successfully"; else echo "Login Failed";
```

- **QUERY RESPONSE:**

Algorithm Used: (True-False) Pattern Matching Algorithm

(Rule-based) Chatbot Algorithm utilized here is Pattern Matching Algorithm. **AIDA** uses pattern matching to produce responses to queries. This technique deployed matching pattern to generate appropriate response from users' questions, which depends on different types of matching such as simple statements or the meaning of enquires. Pattern matching is often referred to as "brute force" as the developer of the system needs to describe every pattern and its response.

The general idea of working of proposed system algorithm is given as follow:

Step 1: Start

Step 2: Read Query

Step 3: Searches Query in database

Step 4: if keyword = Keyword in database return(response)
goto **step 5** else return ("Sorry Message")

Step 5: Give Response for the query then goto **step 6**

Step 6: If want to ask more/another query
goto **step 2**
else
goto **step 7**

Step 7: Stop

The flowchart for above algorithm can be described as follows:

