ASSIGNMENT
DFD- Placement Portal

GROUP-5:-
RITISH BANSAL - 190101076
SURYANSH SINGH - 190101089
ANANT SHANKHDHAR - 190101011
MAYANK CHANDAK - 190101052

1. Introduction:-

Purpose:-
This software design paper gives an overview of the Placement Portal Application's structure and design. It also demonstrates how functionalities of the SRS will be implemented, which gives the reader an appropriate idea of the working of the application.

Scope of the Product and Target Audience:-
The Placement Portal is intended to be used by the students during preparation for placements and internship drives. The Placement Portal System is for the students of IIT-Guwahati to have easy access to important information and resources directly from their seniors who went through the process which will make their intern and placement preparation easier and more convenient.

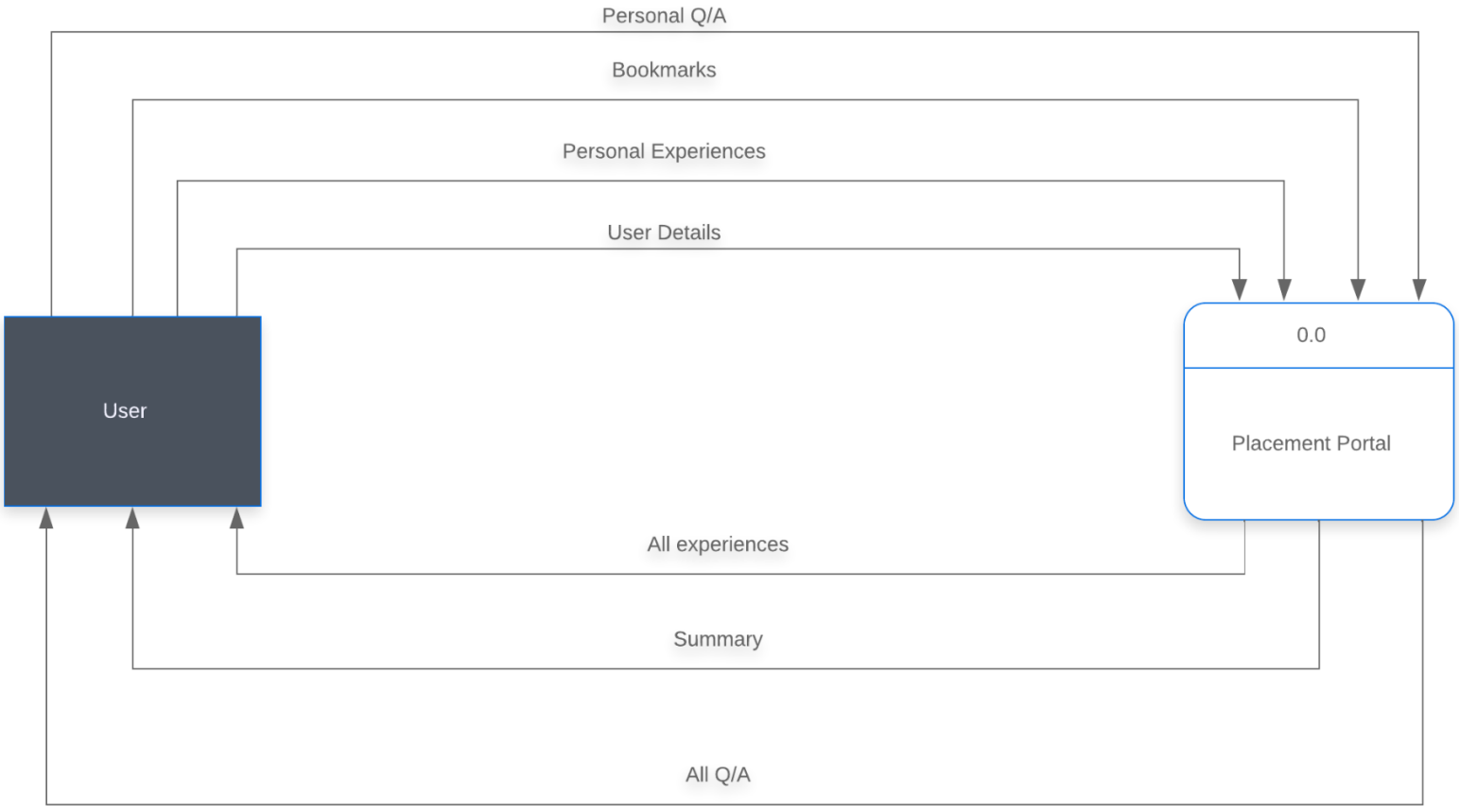
2. System Overview:-

Some of the features which the portal would include are:-

- A facility for the seniors to upload their preparation phase experience and their company working experience
- The current students could see the summarised statistics consolidated company wise, branch wise, year-wise, and role wise.
- We could search through this data based on different criteria, such as skills required, profiles and companies etc
- With the data available we can even add further features such as personalised feeds resource recommendations etc which can be decided later.

3. Data Flow Diagrams

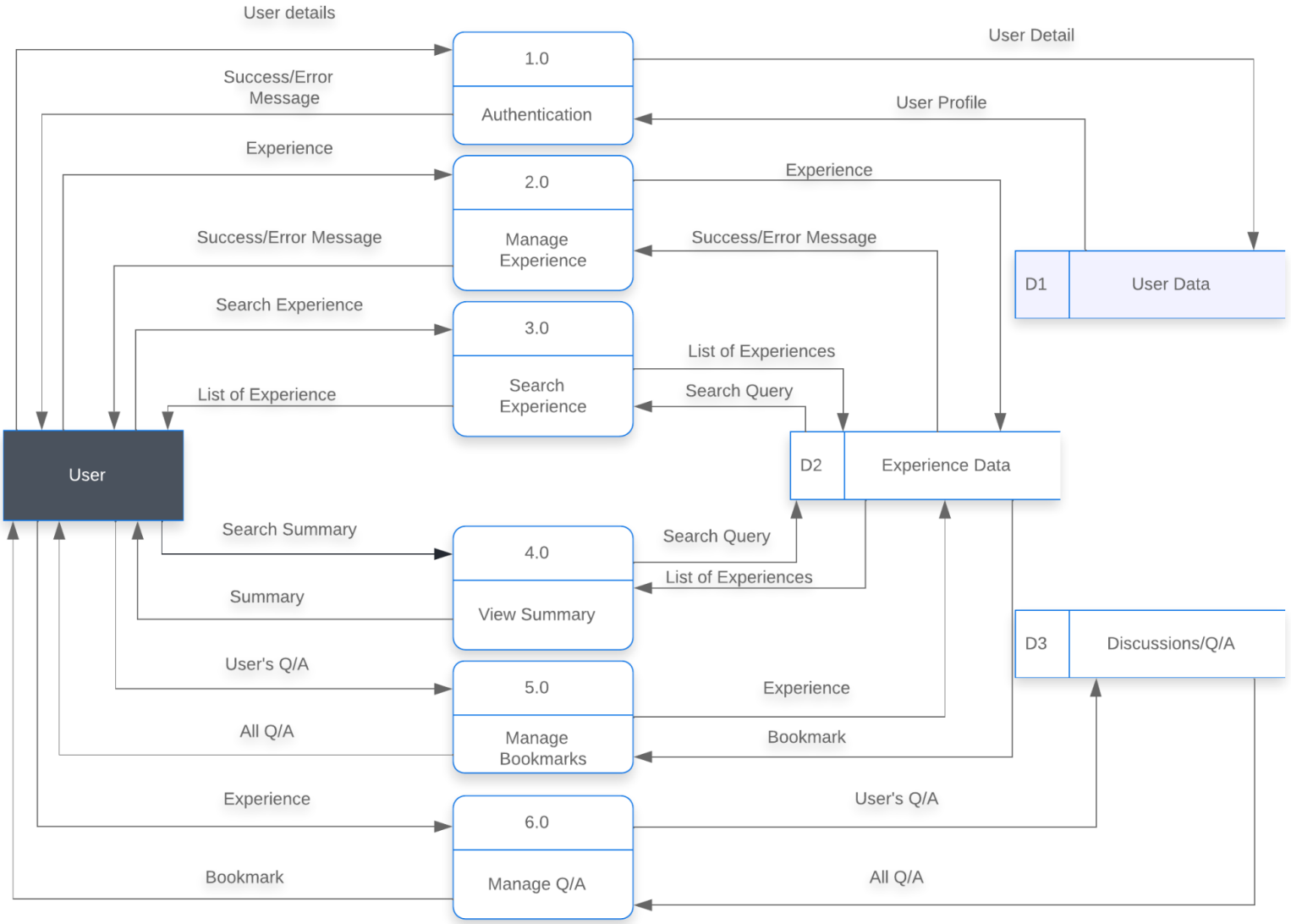
Level 0 Data Flow Diagram(Context Diagram)



Level 1 Data Flow Diagram(Overview Diagram)

There are six main modules:

1. **Authentication:**
   This module deals with the authentication of the user for logging in the user to the application. It involves taking credentials/ details of the user depending on whether the user is signing up or logging in and storing them in the User Data database or verifying them with the entries stored in the database.
2. **Manage experience:**
   This module deals with the experience management of the user. Once the user's session in the application has started, the user can add his/her personal interview experiences on the application for displaying on the common forum. This module manages the addition of all those experiences in a proper format associated with the unique ID of the user (as displayed in the SRS and further levels) and stores them in the Experience Data database. This module also deals with the modification of the experiences if required and updating them in the database.
3. **Search experience:**
   This module deals with returning relevant experiences for a particular query posed by the user. It interacts with the user to get the search query and return the appropriate experiences posted by other users. It also interacts with the Experience Data database to query the given keywords by the user in a proper format and retrieve results from the stored data to return them to the user by displaying them on the application.
4. **View Summary:**
   This module deals with returning the summary for a particular query raised by the user. It fetches appropriate experience results from the database based on the search query by the user and generates a summary based on the results and returns it to the user by displaying it on the application.
5. **Manage bookmarks**
   This module deals with managing bookmarked user experiences by a particular user. The module stores and fetches the information related to bookmarked experiences for all the users and also the like/dislike related information for user profiles. It returns the bookmarked experiences for a user when requested and also updates the list when added by a user for a profile.
6. **Manage Q/A**
   This module deals with managing questions and answers of all the users on the application. It is responsible for saving a user's posted question or posted answer for a particular question. It is also responsible for retrieving the question-answer instances to display on the common forum when requested by the user.


User details
1.0
Authentication
User Detail
Success/Error Message
User Profile
Experience
2.0
Manage Experience
Experience
Success/Error Message
Success/Error Message
D1
User Data
Search Experience
3.0
Search Experience
List of Experiences
List of Experience
Search Query
User
D2
Experience Data
Search Summary
4.0
View Summary
Search Query
List of Experiences
Summary
D3
Discussions/Q/A
User's Q/A
5.0
Manage Bookmarks
Experience
All Q/A
Bookmark
Experience
6.0
Manage Q/A
User's Q/A
Bookmark
All Q/A

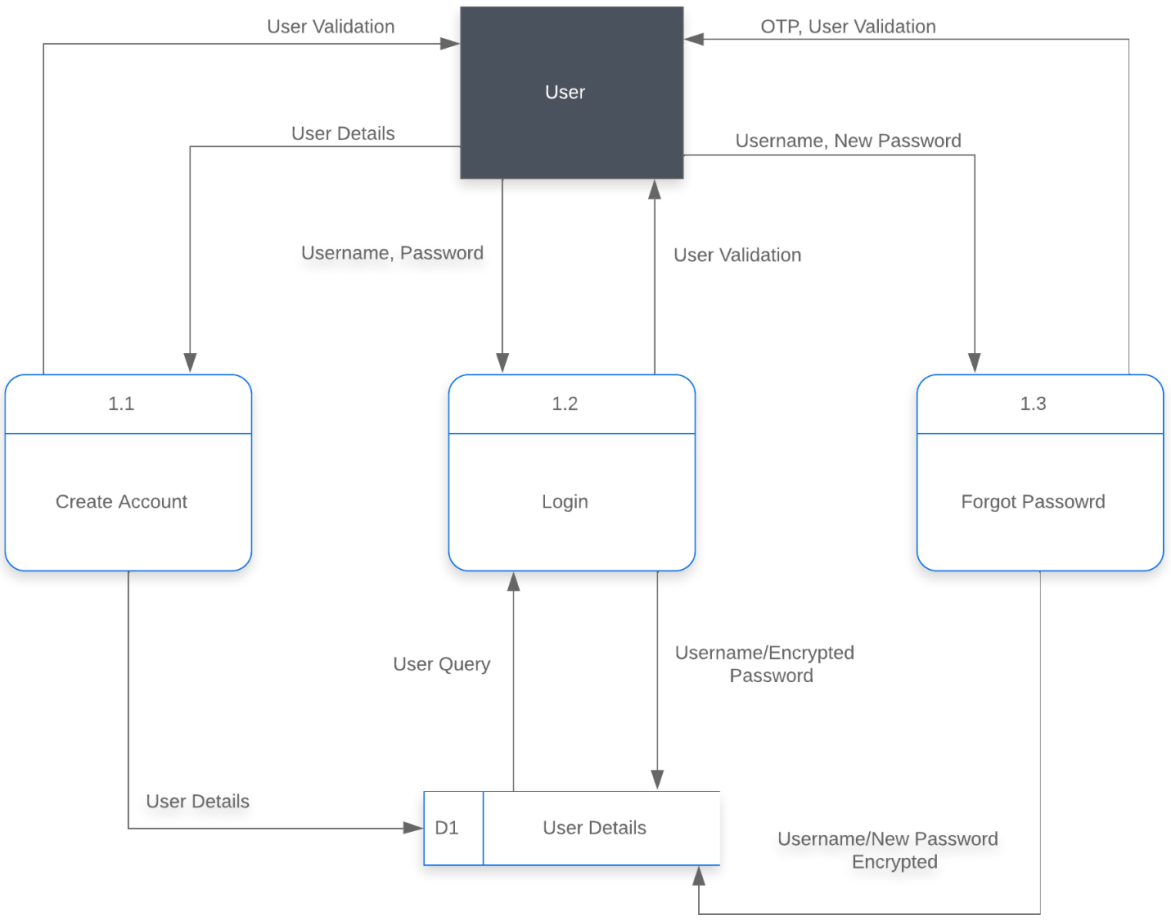
Module 1- Authentication

There are three processes in this module:-

1. **Create Account:-** This process performs the function of registering a new user to the platform. The username, email id and password are taken as input from the user. On the execution of this process, a new entry is created in the User Details data store and the user receives confirmation in the form of a success/error message.

2. **Login:-** This process performs the function of authenticating an existing user. The username and password are taken as input from the existing user. On the execution of this process, the input is verified with the User Details data store. If the username and password are correct then the user receives a success message and is given access to his/her account on the platform. Otherwise, they receive an error message.

3. **Forgot Password:-** This process performs the function of setting a new password for a user who forgot their login credentials. The username, and new password are taken as input from the user. During the execution of this process, the user receives a one time password(OTP) and a validation message on their email id. If the validation succeeds then the password corresponding to the username in the User Details data store is changed to the new password.

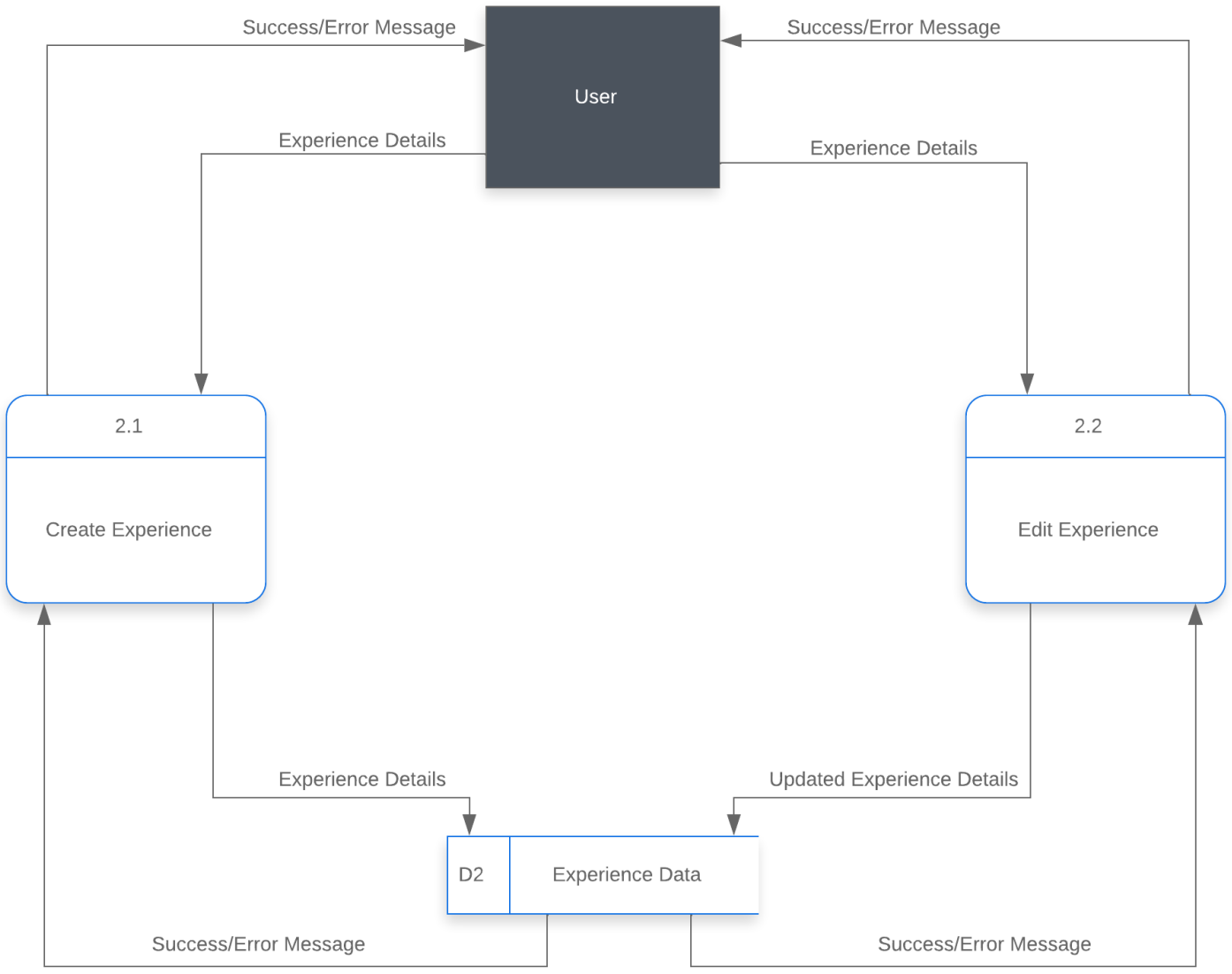


Module 2- Manage Experience

There are 2 main processes in this module:-

1. **Create Experience:-** This process performs the function of adding a new internship/placement interview experience by taking the details of the experience as input from the user and creating a new experience entry in the Experience Data (data store) containing these details.

2.**Edit Experience:-** This process performs the function of editing/deleting an internship/placement interview experience . The experience id, type of operation and changed details are taken input from the user. The experience corresponding to the id is changed in the Experience Data (data store)



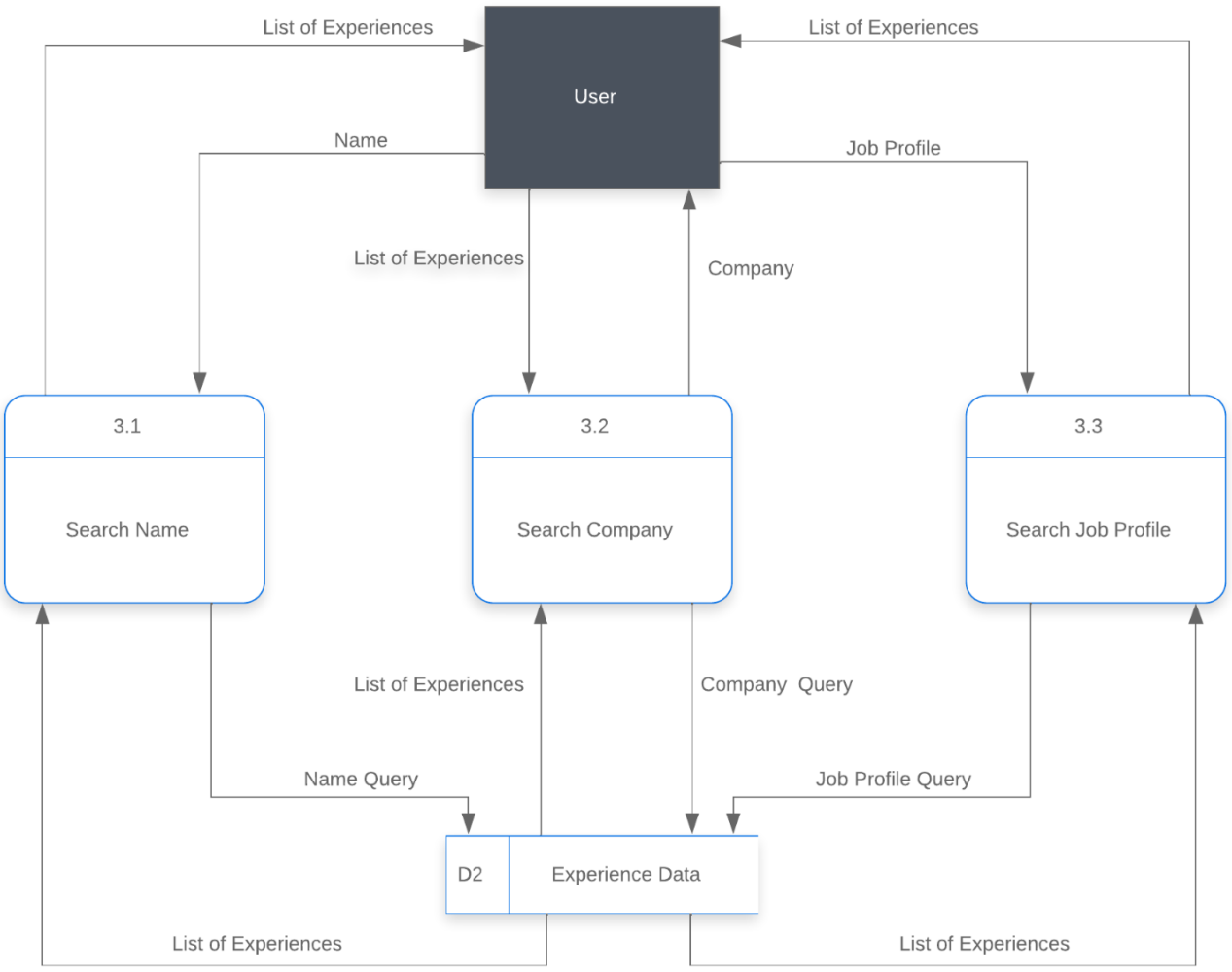
Module 3- Search Experience

There are 3 main processes in this module:-

1.**Search Name:-** This process takes the name of a person from the user. The process then searches for all the experiences added by the person in the input search query from the Experience Data (data source) and returns a list of experiences filled by the input person to the user.

2.**Search Company:-** This process takes the name of a company from the user. The process then searches for all the experiences belonging to the search query company from the Experience Data (data source) and returns a list of experiences corresponding to the searched company to the user.

3.**Search Job Profile:-** This process takes the name of a job profile from the user. The process then searches for all the experiences belonging to the search query job profile from the Experience Data (data source) and returns a list of experiences corresponding to the searched job profile to the user.



Module 4- View Summary

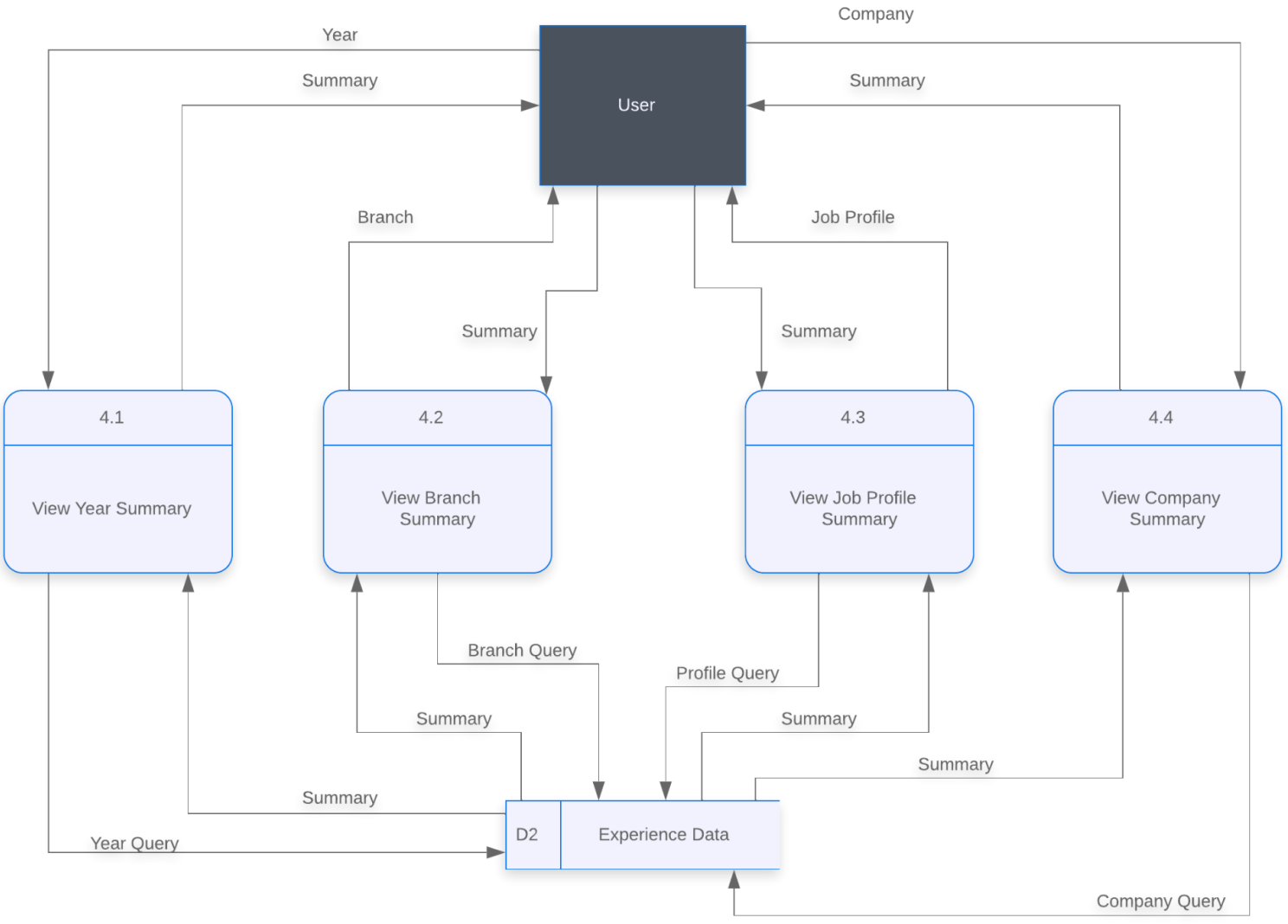
There are 4 main processes in this module :-

1.**View Year Summary:-** This process takes the year input from the user, looks for all experiences belonging to that year from Experience Data (data source) and returns the summary of all experiences from that year to the user.

2.**View Branch Summary:-** This process takes the branch name input from the user, looks for all experiences belonging to that branch from Experience Data (data source) and returns the summary of all experiences from that branch to the user.

3.**View Job Profile Summary:-** This process takes the Job Profile input from the user, looks for all experiences belonging to that Job Profile from Experience Data (data source) and returns the summary of all experiences from that Job Profile to the user.

4.**View Company Summary:-** This process takes the company input from the user, looks for all experiences belonging to that company from Experience Data (data source) and returns the summary of all experiences from that company to the user.



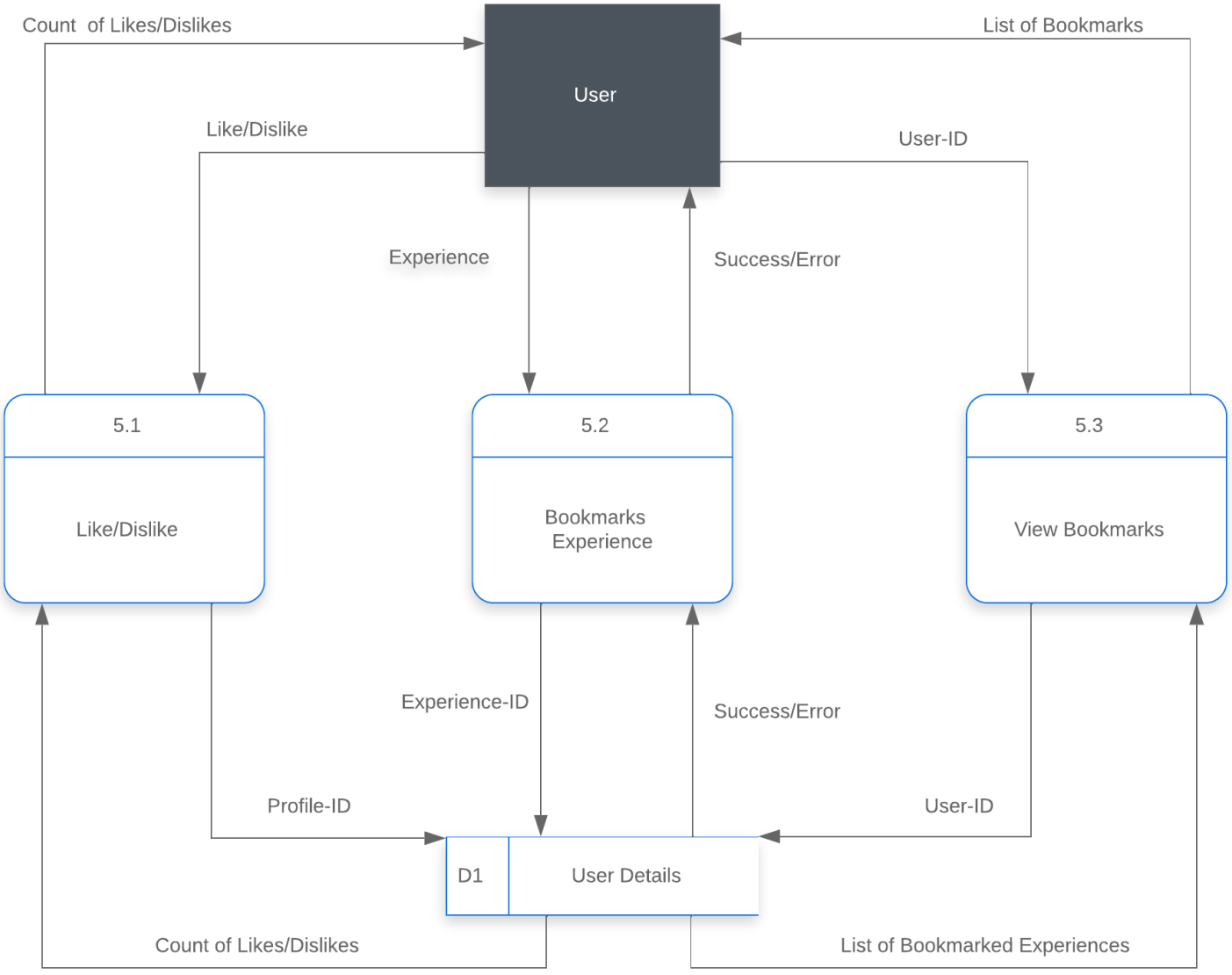
Module - 5 - Manage Bookmarks

There are three processes in this module:-

1. **Like/Dislike**: This process takes the profile and whether the user likes it or dislikes it and then stores the same in User Data. It then returns the number of likes and dislikes on that profile to the user.

2. **Bookmarks Experience**: This process takes the experience to be bookmarked by the user and stores it in User Data. It then returns a success/error message to the user.

3. **View Bookmarks**: This process takes the user ID of the user and returns the List of experiences bookmarked by the user.



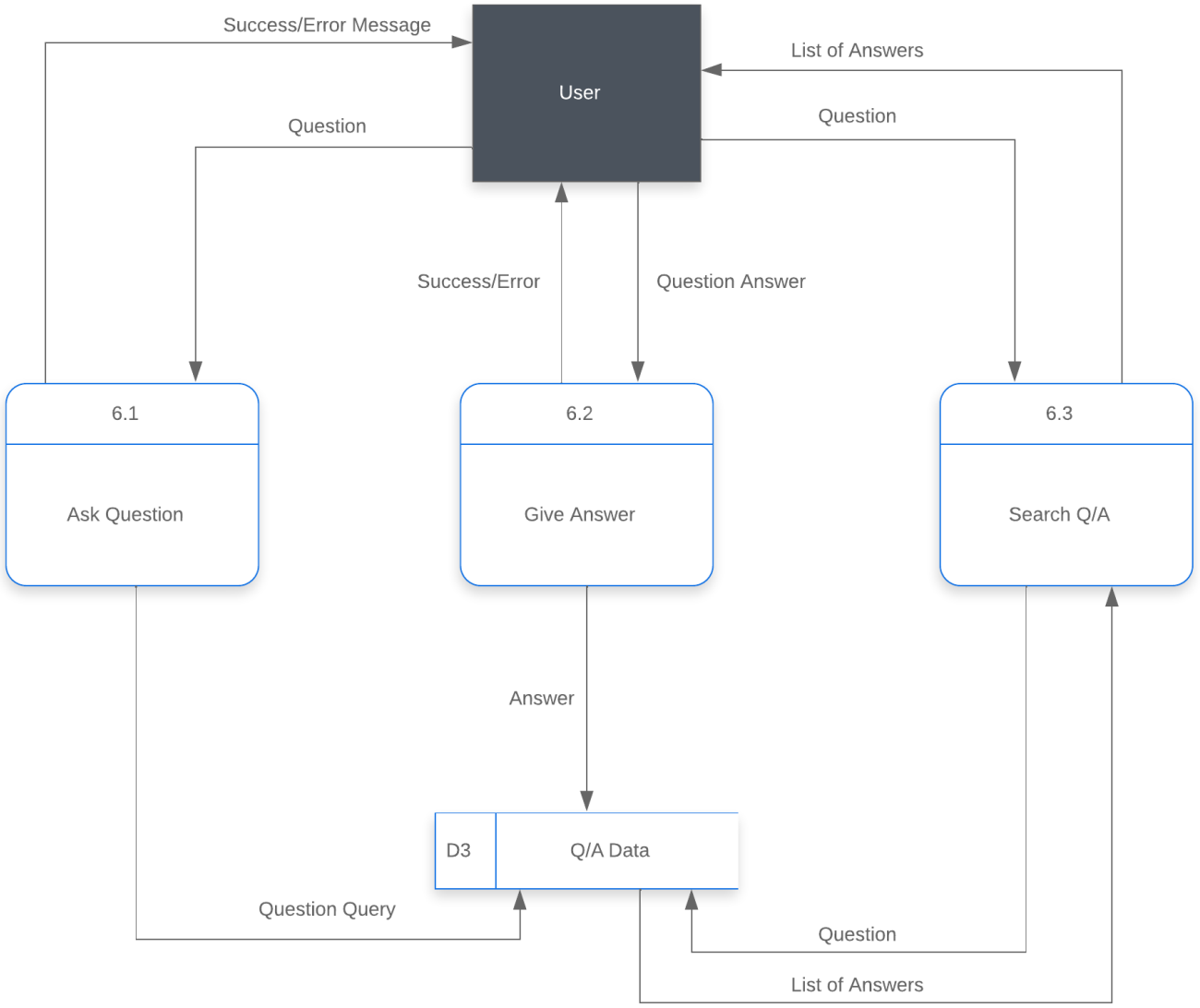
Module - 6 - Manage Q/A

There are three processes in this module:-

1. **Ask Question**: This process takes the question from the user and stores it in Q/A Data and returns whether the process was successful or not.

2. **Give Answer** : This takes the answer and the question-ID of the corresponding question from the user and stores them in Q/A Data and returns whether the process was successful or not.

3. **Search Q/A** : This process takes the question for which the user wants answers and searches it in Q/A Data and returns the answers found corresponding to the searched question.process



Data Design

1. Data Description(ER Diagram)

The following entity-relationship diagram graphically illustrates how data would be stored and used in the application. The rectangles represent the actual data stores (entity sets), the ellipses attached to each data store represent its attributes that characterise it, and the dia-monds describe the relationship between data stores. This entity-relationship diagram follows the standard ER diagram conventions

Data Dictionary

1. **User Data**
   It stores the credentials of every user who uses this application. It also stores public profile, bookmarks and likes/dislikes of specific users.
   It has the following attributes -
   - **Credentials** - This stores username,email-id, and password for each user. The username will be unique for all users.
   - **Public Profile** - This store's various information about a specific user. This includes Branch, Graduation Year, Name, Programme, Social Profiles.
   - **Bookmarks** - This is a multivalued field. Each user can bookmark multiple experiences and this stores all the bookmarked profiles of the user.
   - **Like/Dislike** - This stores the list of profiles that users have liked and disliked.

2. **Experience**
   It stores all details and all information related to experiences which is the basic structural component of the application. Every user can create multiple experiences. It has following attributes -
   - **Experience ID** - uniquely identifies the experience.
   - **Experience Type** - Stores type of experience as of internship or placement from the user.
   - **Resources Used** - Stores the resources associated with this experience.
   - **Suggested Tips** - Stores the tips associated with this experience given by the user.
   - **Company** - Stores the name of a company which the experience is associated with.
   - **EffortPut** - Stores the efforts given by the user to in the form of hours and days.
   - **Resume -** Stores the resume if provided by a user given to a specific company.
   - **Experience of different rounds** - Stores the text experience of different rounds. This will be a multivalued field as user can provide many experiences for different number of rounds.
   - **Recruitment Process** - Stores the information about how recruitment is taken and what are special things about this company.

3. **Discussion**

   It stores all the question answer sets posted by and modified by different users. A question can be posted by any user and each question can have multiple answers by different users.

   It has following attributes -

   - **QuesID** - This uniquely identifies the question.
   - **Question** - Stores the question asked by the user in the form of a string.
   - **Answers** - This is a multivalued field. Each question can have multiple answers. It stores a list of answers for the question.

---

5. Design Justification

We should try to achieve high cohesion within a module and low coupling between different modules. And, we have tried to do the same. We have provided a summary of the different types of cohesion and coupling found in our design and listed a few examples of the same. Also, we have tried to justify the reasons for the same.

a. Cohesion

- **Logical Cohesion:** It is found if all the functions are performing similar operations. Almost all our modules have logical cohesion within them. The different modules we created are almost independent of each other and take care of different entities and provide different functionalities. e.g. Manage Experience is related to adding and editing experiences. Manage Q/A is related to managing questions and answers only.
- **Temporal Cohesion**: It is found if all the functions are used in almost similar time span. This type of cohesion was found in some of our modules. e.g. Manage Experience has functions - Create Experience which would have several functions within it for getting the experience from the user. We see temporal cohesion here, which would be executed in almost similar time frame.
- **Functional Cohesion:** If the functions are part of the same procedure, we find function cohesion. We could find function cohesion in several of our modules. e.g. Authentication module: all the functions in this module are part of signing up user or signing in user or modifying the credentials of a user.
- **Communication Cohesion:** If all the functions refer to or update the same data structure, we find communication cohesion. We found communication cohesion in several of our modules. e.g. Manage Experience: this module is using Experience Data database almost all of the time. This module exclusively updates its data structure every time.

b. Coupling

- **Data Coupling:** Data Coupling happens if two modules communicate through a data item. This type of coupling is almost not happening in our design.
- **Control Coupling:** Control Coupling happens if data from one module is used to control the flow of instruction of another module.

For eg, Authentication module has coupling with all other modules because at each level credentials of the user will be checked.

- **Content Coupling**: This happens when two modules share the same code. This type of coupling is not happening in our design.