Software Requirement Specification for Contact Management

Name	**Monisa K**
Roll no	**7376222BT162**
Seat no	**157**
Project id	**28**
Problem statement	**Contact Management**

1. Problem Statement:

The hiring data and industrial visit contacts are currently managed using disorganised ways (spreadsheets, emails, and notes). This results in a loss of information, inefficiency, and trouble following up with prospects. To facilitate industrial visit coordination and streamline the hiring process, we require a centralised application that stores and arranges contact details, visit reports, and candidate information.

2. Project Flow:

- **Collect Requirements:** Have a meeting with stakeholders to find out how they presently handle employment information and contacts. Determine their requirements and areas of discomfort.
- **Describe the data model:** Make a list of all the data you wish to keep, including contact information, appointment schedules, business details, and applicant qualifications.
- **Identification of Entity and Attribute:** Divide the data model into its fundamental components, such as Contacts, Companies, Visits, and Candidates. Describe each entity's attributes (information) (e.g., name, email, visit report, resume).
- **Keys & Relationships:** Determine the relationships between entities (e.g., a Contact may attend several Visits). Establish a distinct identity (Primary Key) for every table (Contact ID, for example).

- **Design with logic:** To represent entities, properties, and relationships visually, create an entity-relationship diagram (ERD).
- **Normalisation:** Optimise the design by reducing redundant data and increasing efficiency (e.g., by deleting recurring groupings).
- **Physical Design:** Convert the logical model to the particular database platform (PostgreSQL, MySQL, etc.) that you plan to utilise. For every attribute, specify the data type (text, date, etc.).
- **Database Creation:** Based on the physical design, create the database structure using the selected platform.
- **Validation and Testing:** Add example data to the database and test adding, editing, and retrieving information functions.
- **Documentation:** For future reference, keep a record of the whole database design process, including the ERD and data dictionary.

3. Functional Requirements:

Contact Management:

- Add, edit, and delete contact details (name, email, organization, etc.).
- Organize contacts by category (e.g., industry, university).
- Search and filter contacts based on various criteria.

Industrial Visit Management:

- Schedule and track industrial visits.
- Store and manage visit reports, including notes and takeaways.
- Link visit reports to relevant contacts and companies.

Hiring Management:

- Create and manage job postings for positions arising from visits.
- Store and manage candidate applications (resumes, cover letters).
- Track candidate progress through the hiring pipeline.

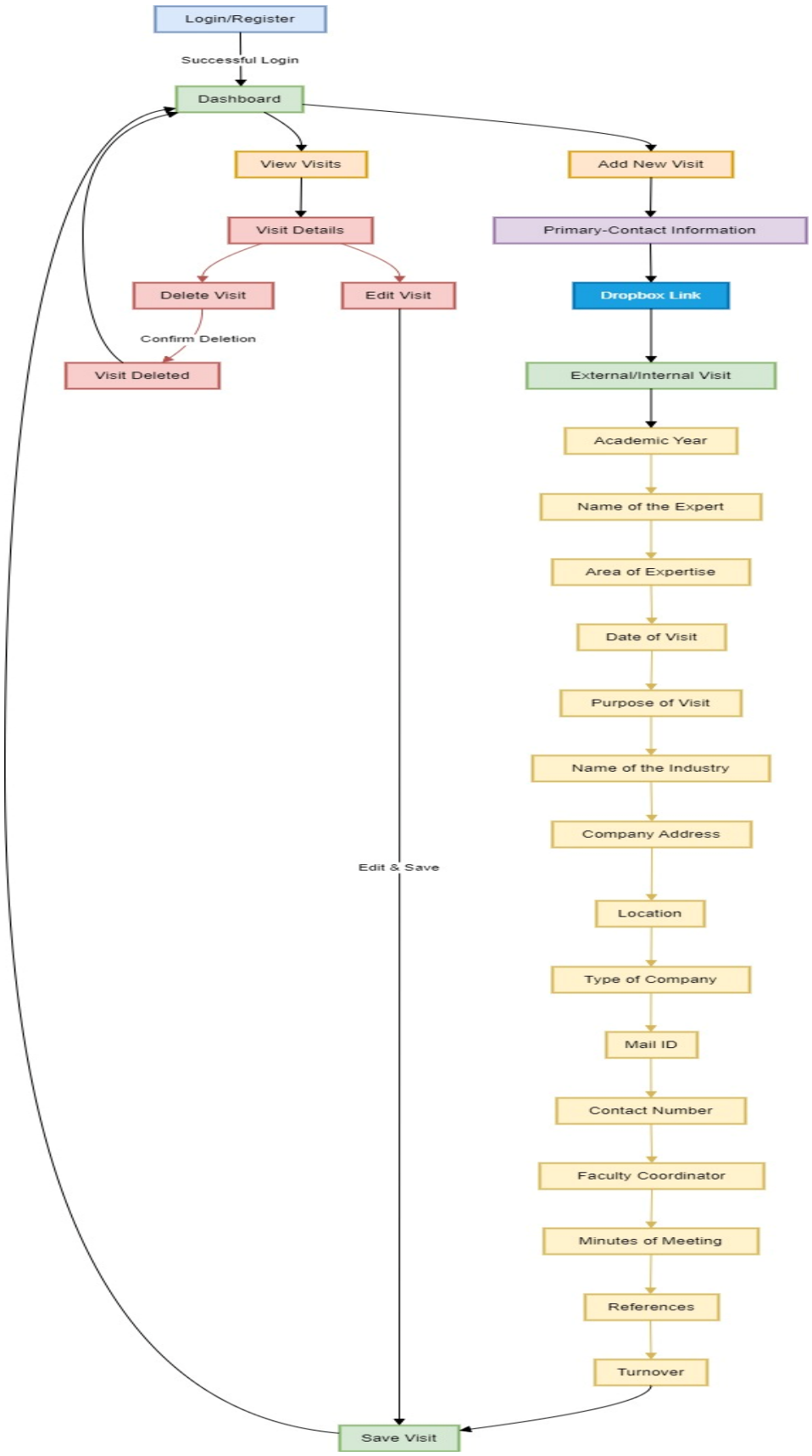
4. Non-Functional Requirements:

- **Security:** Protect user identity and permission to access data. Put safeguards in place to stop illegal access and data breaches.
- **Performance:** Even with a lot of data, the app should be responsive and load quickly.
- Data retrieval and queries must to be effective.
- **Scalability:** An expanding user, contact, or visitation count should not be a problem for the database.
- **Usability:** All parties involved in the app—faculty, students, and industry personnel—should find the interface easy to use and intuitive.
- Give users precise instructions and direction on how to use the features of the app.
- **Availability:** There should be little downtime and constant access to the app.
- Establish a plan for data backup and recovery in the event that the system fails.

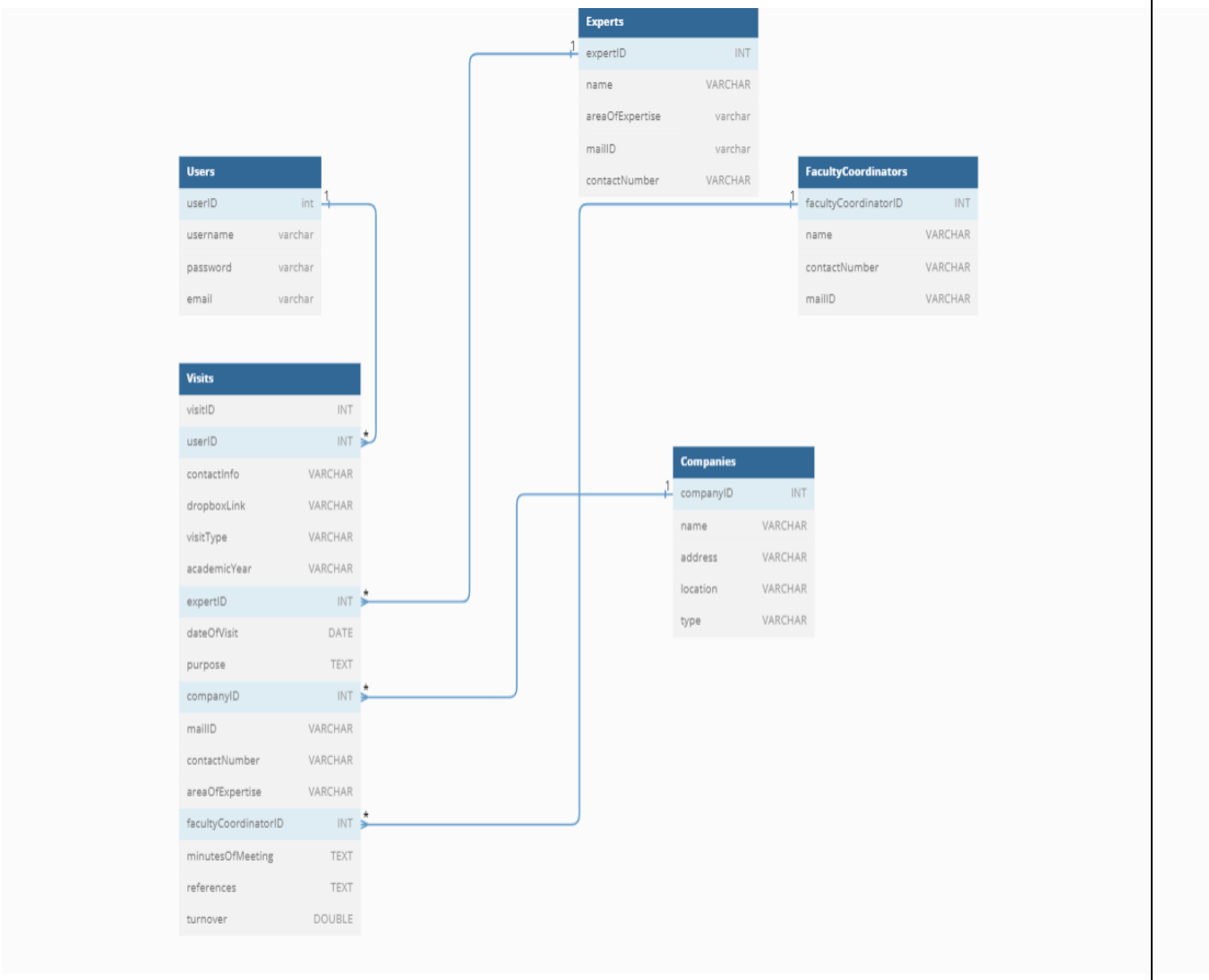
5. Technical Requirements:

Front End	• React (JS Library for building user interfaces)
Back End	• Node.js with Express.js
Database	• MongoDB(NOSQL Database)
API	• OpenAPI

6. Flowchart:


Login/Register
Successful Login
Dashboard
View Visits
Add New Visit
Visit Details
Primary-Contact Information
Delete Visit
Edit Visit
Dropbox Link
Confirm Deletion
Visit Deleted
External/Internal Visit
Academic Year
Name of the Expert
Area of Expertise
Date of Visit
Purpose of Visit
Name of the Industry
Company Address
Edit & Save
Location
Type of Company
Mail ID
Contact Number
Faculty Coordinator
Minutes of Meeting
References
Turnover
Save Visit

7. ER diagram:


Experts
expertID INT
name VARCHAR
areaOfExpertise varchar
mailID varchar
contactNumber VARCHAR
Users
userID int
username varchar
password varchar
email varchar
FacultyCoordinators
facultyCoordinatorID INT
name VARCHAR
contactNumber VARCHAR
mailID VARCHAR
Visits
visitID INT
userID INT
contactInfo VARCHAR
dropboxLink VARCHAR
visitType VARCHAR
academicYear VARCHAR
expertID INT
dateOfVisit DATE
purpose TEXT
companyID INT
mailID VARCHAR
contactNumber VARCHAR
areaOfExpertise VARCHAR
facultyCoordinatorID INT
minutesOfMeeting TEXT
references TEXT
turnover DOUBLE
Companies
companyID INT
name VARCHAR
address VARCHAR
location VARCHAR
type VARCHAR