Database Design Document

for

Lost and Found Information System

Prepared by

Group 10

Rahul B Menon	B210482CS	rahul_b210482cs@nitc.ac.in
Pranav Prashant	B210460CS	pranav_b210460cs@nitc.ac.in
S Rishi Mohan	B210552CS	rishi_b210552cs@nitc.ac.in
Rahul P Aroli	B210544CS	rahul_b210544cs@nitc.ac.in
Ankesh Kumar	B200821CS	ankesh_b200821cs@nitc.ac.in

INSTRUCTOR:	Dr. M Prabu and Dr. Abdul Nazeer K
COURSE:	Database Management Systems
LAB SECTION:	CS04
DATE:	6/11/2024

CONTENT

1 Purpose

The Database Design Document (DDD) for a Lost and Found Information System (LFIS) is crucial for several reasons. It acts as a comprehensive reference, providing a clear and structured outline of the system's database design to all project stakeholders, ensuring that everyone involved understands and aligns with the database's structure and functionality. The DDD documents specific system requirements, serving as a blueprint for developers and enabling efficient, quality-assured development. In essence, the DDD plays a pivotal role in guiding the design and development of the database for the Lost and Found Information System, enhancing communication, project efficiency, and quality assurance. An Entity-Relationship model is created and is converted to a relational schema of the target Database Management System(DBMS).

1.1 Document Objectives

The Database Design Document has the following objectives:

- **System-understanding:** Facilitate a clear and thorough understanding of the Lost and Found Information System's database among all stakeholders, including developers, project managers, and end-users, by offering a comprehensive overview of its structure and operations.

- **Software Design and Database Specification:** Provide a detailed outline of the software design and specifications for the Lost and Found Information System database, as well as the system's architecture and accessible components available to both users and system developers through a Database Management System (DBMS).

- **Data Organization:** Define how lost and found data is organized within the database, including the types of data to be collected, stored, and managed, ensuring that it aligns with the system's requirements.

- **Documentation:** Create a reference guide that documents the database structure, data dictionaries, and any additional information necessary for database administration and future maintenance.

- **Foundational Methodology for Database Implementation:** To establish a foundational methodology for implementing the database and associated software components, thereby assisting in the extraction of essential details required for developing the application.

1.2 Intended Audience and Document Overview

This document is intended to serve several group of audience members like :

- **Instructors/Graders:** This document caters to the college instructors or graders who evaluate the project's design and database implementation.

- **Team Members/Developers**: To understand and reference this document during the development phase.

- **Future Developers:** For maintenance or enhancement of the system in the future. Enhances future developers to understand the database's structure and logic.

- **Project Stakeholders:** Abilitates the project stakeholders to grasp the project's scope and purpose from this document.

- **Quality Assurance personnel:** for testing and validating the requirements given in this document.

The following section of the document, titled **"Assumptions and Constraints"**, provides an overview of the assumptions made and the limitations set during the product development. The subsequent section, **"Database Wide Design"**, focuses on elucidating the system's behavior, emphasizing key roles and actions, and delving into specifics regarding the DBMS platform, security prerequisites, performance considerations, and availability decisions. In the fourth section, **"Database Administrative Functions"**, there is the Entity Relationship Model, the relational schema derived from the ER diagram, as well as information regarding normalization and data format specifics.

1.3 Definitions, Acronyms and Abbreviations

- **DDD:** Database Design Document
  A comprehensive document that outlines the structure, organization, and specifications of a database system.

- **LFIS:** Lost and Found Information System
  A software system designed to manage and track lost and found items within the college providing a platform for reporting, cataloging and recovering lost items.

- **SRS:** Software Requirements Specification
  A document that describes what the software will do and how it will be expected to perform.

- **DBMS:** Database Management System
  It is a software application or system that facilitates the creation, management and manipulation of databases.

- **MySQL:** An open source relational database management system.

- **ReactJS:** A frontend development framework written in JavaScript developed by Facebook.
- **Database:** Stores the catalog of the books available in the library and details of the users.
- **NITC:** National Institute of Technology, Calicut
- **1NF:** First Normal Form
- **2NF:** Second Normal Formal
- **3NF:** Third Normal Form

1.4 References and Acknowledgments

- R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7/e, Pearson Education, 2016
- R. Ramakrishnan and J. Gehrke, DatabaseManagement Systems,3/e, McGraw Hill, 2003.
- P. Rob and C. Coronel, Database SystemsDesign, Implementation and Management, 7/e, Cengage Learning, 2007.
- Fundamentals of Database Learning by Ramez Elmasri
- http://www.sdlcforms.com/PopupForm-DatabaseDesignDocument.html
- https://app.diagrams.net/
- https://creately.com/
- wikipedia

2 Assumptions and Constraints

2.1 Assumptions

The following are the assumptions made while developing this product :

- All the users are from the NITC community.
- Limited User Roles: Assume that there are only a few user roles, such as administrators, users who report lost items, and users who report found items. You can simplify by not considering advanced user roles or permission systems.
- The system must have storage capacity and render fast access to the database.
- The system will be available at all times
- The users know English as the interface will be entirely in English.
- Assume that item descriptions and details are text-based. More complex data types like images or documents for item descriptions can be disregarded.
- Since the application is web-based, there is a need for an internet browser. It will be assumed that users will possess decent internet connectivity.
- It is also assumed that the user is familiar with an internet browser and handling the keyboard and mouse.

2.2 Constraints

The following design and implementation constraints are employed in the system:

- The system may have constraints on the volume of data it can handle, including the number of items, users, and records.
- Limited timeframe for development, which can affect the depth and complexity of the features that can be implemented.
- Administrative access is protected with exclusive credentials.
- There may be constraints related to security, including data encryption, user authentication, and access control measures that must be implemented.
- Constraints on how the system should scale to accommodate a growing user base or an increasing number of lost and found items.

3 Database- Wide Design Decisions

3.1 Behavior

3.1.1 User Authentication/ Verification

- Authorized system users (students and employees) are granted access by entering their login credentials, comprising a unique login ID and password.

- New users seeking access are required to complete the registration process, which necessitates the creation of a new account by the submission of their respective Roll Number or Employee ID.

- A prerequisite for system access is that the users must belong to the NITC community.

- Following successful registration, users are automatically redirected to the subsequent page in the system.

3.1.2 User Roles

Users, after logging into the system is presented with the following options:

- Lost an Item.
- Found an Item.
- Admin.

3.1.2.1 Lost An Item

The user is presented with the option to search the lost and found item list for the lost item. In the event that the user finds his item within the listing, he is provided with an option to submit a formal request for its retrieval. If the user is unable to locate his item within the listing, he can officially designate his item as lost and provide a comprehensive description about it.

3.1.2.2 Found an Item

The user is provided an option to report found items in the item listing by providing a detailed description of the item, specifying where it was found and indicating the date of discovery. Should an individual come across an item that has been in the lost item listing, they have the option to officially designate the item as found, and provide details of discovery.

3.1.2.3 Admin

Admin, once signed in with proper credentials, is presented with the following options:

- Manage the users including deletion.
- Delete an item from the item listing.
- View transaction history of all users/items.

The admin acts as the super-admin of the system who can update the database. The admin role is highly secure and is limited to a very few users.

3.2 DBMS Platform

LFIS is a web application that provides the user with a clear and interactive experience. The design is simple, and all the interfaces follow a standard template. The web application is expected to work on web browsers. The application allows users to log into the system with corresponding credentials and is directed to different pages according to their roles. The functionalities extended to different users differ in accordance with their roles.

3.3 Security Requirements

The system will store all the data in a secure database. The interacting users will be able to view information but will not have the privilege to modify/edit it. This privilege will be given only to the admin, and only they have the right to update the database. Admin oversees user accounts and item listings.

After removal of an item from the item listing, the system retains the information about the item, along with the details of the individual who retrieved it, for a specified duration of time.

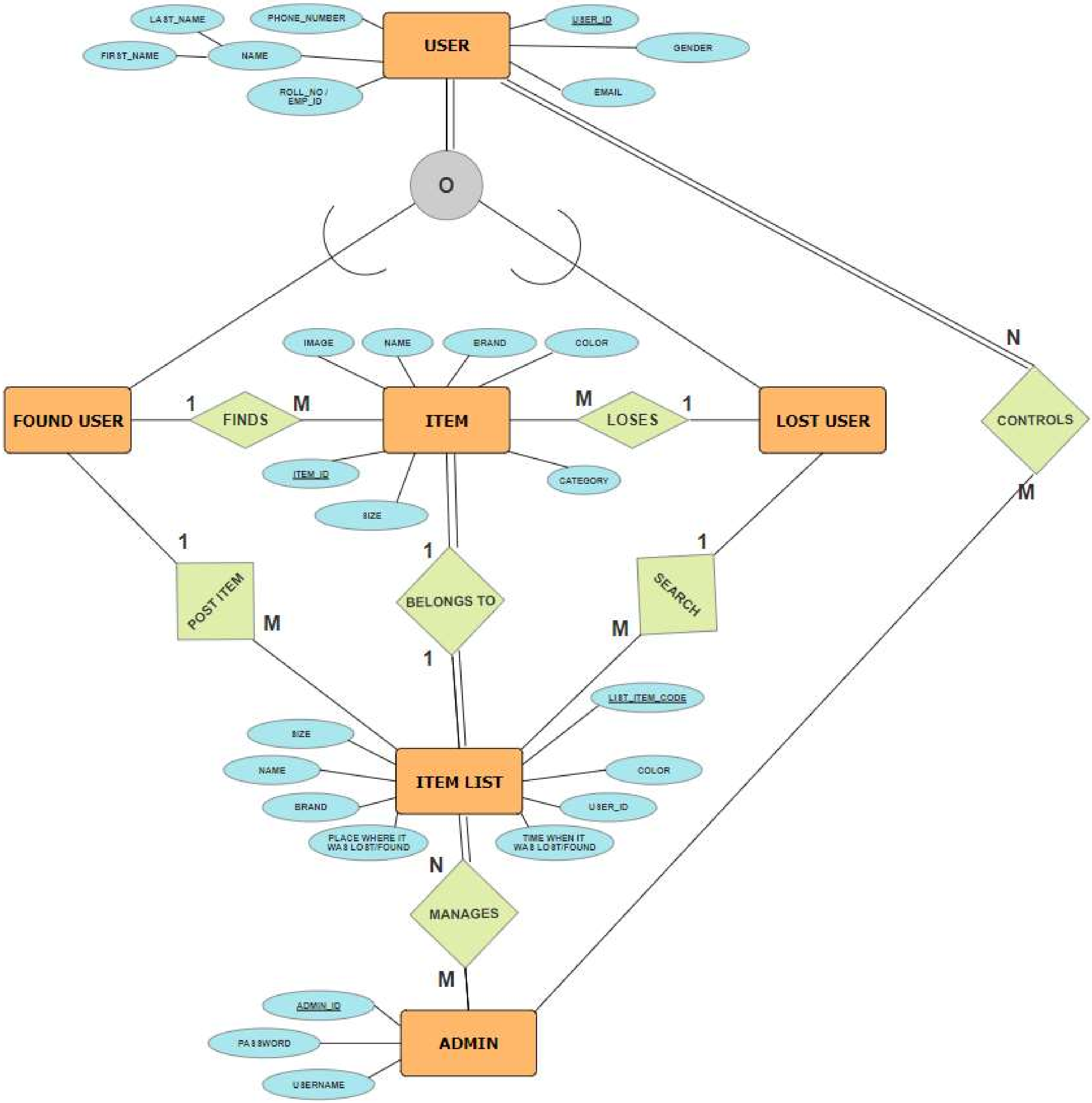
In terms of the safety aspect, the system does not pose a threat to its users. To combat attacks by malware, backing up the database is advised.

3.4 Performance and Availability Decisions

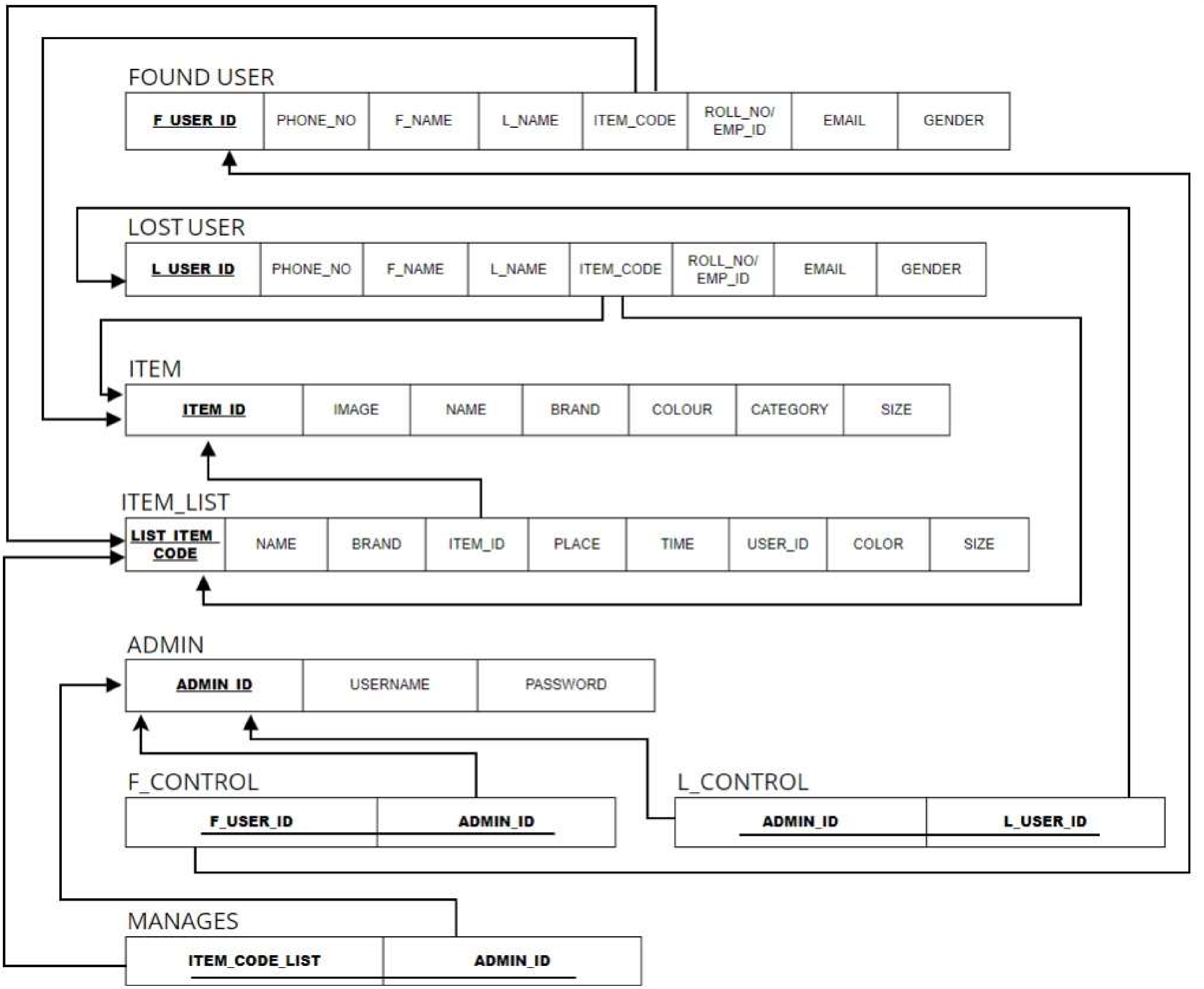
The search retrievals depend upon the number of items in the item listing and the updates made to the system. This system is designed to interact with users (students and employees) across the campus.
The system will respond to the user in less than a second of submitting a request. The view of history may take a few seconds extra. Overall, the performance will be fast and accurate.
The system will be capable of handling a large amount of data and hence accommodate a high number of items, user transaction history, user credentials, etc.
The system is scalable to a certain extent and can accommodate a potential increase in the number of users and items.

4 Database Administrative Functions

4.1 Entity-Relation Model



4.2 Relational Schema



FOUND USER

F USER ID	PHONE_NO	F_NAME	L_NAME	ITEM_CODE	ROLL_NO/ EMP_ID	EMAIL	GENDER

LOST USER

L USER ID	PHONE_NO	F_NAME	L_NAME	ITEM_CODE	ROLL_NO/ EMP_ID	EMAIL	GENDER

ITEM

ITEM ID	IMAGE	NAME	BRAND	COLOUR	CATEGORY	SIZE

ITEM_LIST

LIST ITEM CODE	NAME	BRAND	ITEM_ID	PLACE	TIME	USER_ID	COLOR	SIZE

ADMIN

ADMIN ID	USERNAME	PASSWORD

F_CONTROL

F_USER_ID	**ADMIN_ID**

L_CONTROL

ADMIN_ID	**L_USER_ID**

MANAGES

ITEM_CODE_LIST	**ADMIN_ID**

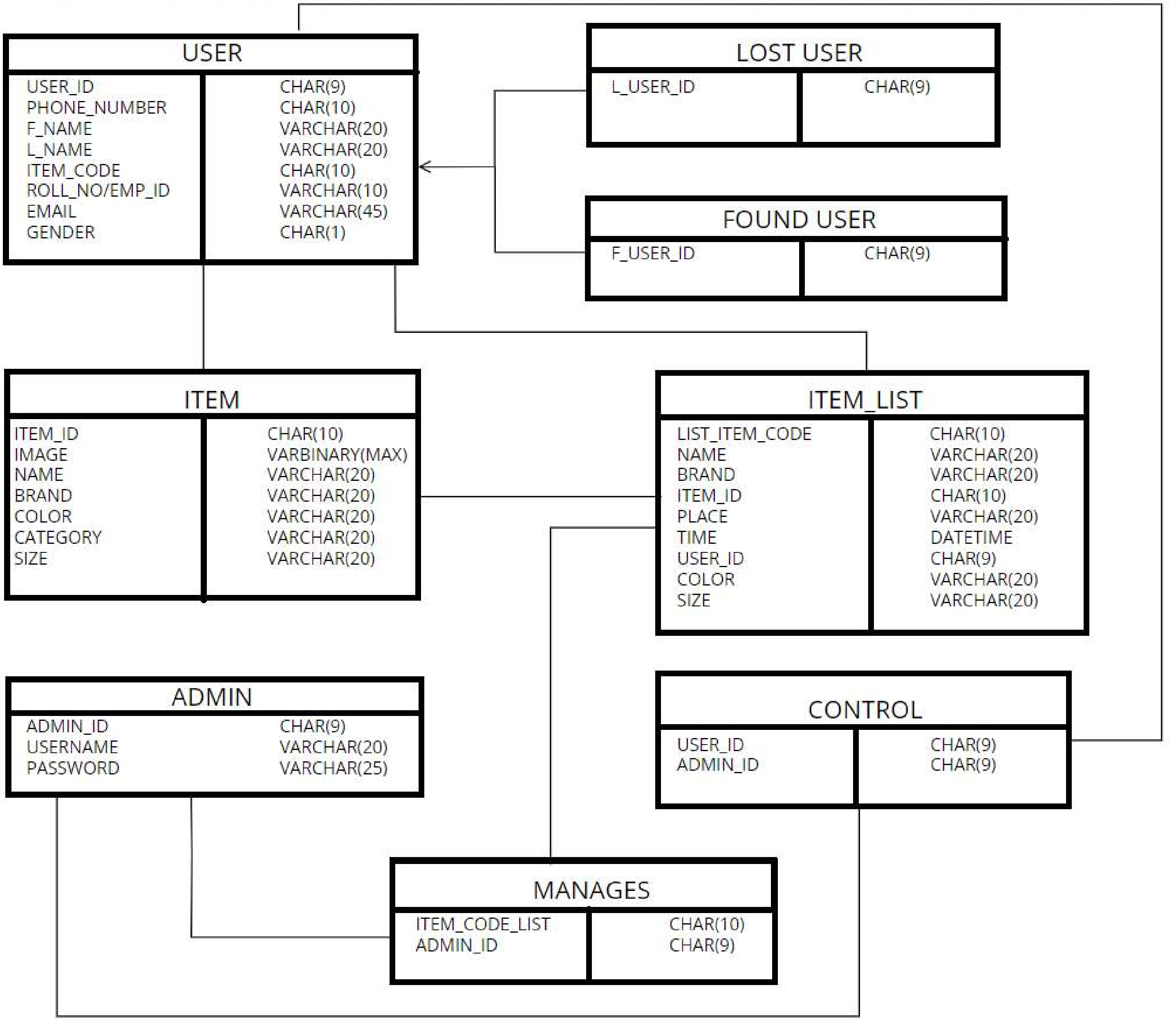
4.3 Normalization

- **1NF-**
    - ➢ The tables are in 1NF, as there are no multivalued or composite attributes.
    - ➢ Each table in the database has a primary key that uniquely identifies each record. For example, in the "Items" table, "Item_ID" serves as the primary key. This ensures that each item record can be uniquely identified by its Item_ID.
    - ➢ Each attribute in the tables contains atomic values. This means that data is not stored as complex, multi-valued, or composite attributes. For instance, in the "Items" table, attributes such as "Name", "Brand", "Color", "Category", and "Item_ID" all contain single, non-divisible values. There are no arrays, lists, or composite data types.
    - ➢ Each record is unique. There are no repeating groups of attributes within a single record. In other words, each record in a table does not contain multiple occurrences of similar data. For example, there are no multiple "Location" values within a single "Item List" record.
    - ➢ Hence the database is 1NF normalized.

- **2NF-**
    - ➢ The tables are already in 1NF as proved above.
    - ➢ There are no **partial dependencies**, that is, there are no non-primary keys solely dependent on only one part of a primary key in any of the tables. **Non-key attributes should be fully functionally dependent on the entire primary key**. Hence the database is 2NF normalized.
    - ➢ There is no non-key attribute that depends only on one part of the primary key (partial dependency). If for a table, there are two attributes that together make up a primary key, then all other non-key attributes should depend upon both of them.
    - ➢ By adhering to 2NF, data integrity is maintained.

- **3NF-**
    - ➢ The tables are already in 2NF as proved above.
    - ➢ There are no transitive functional dependencies in the schema. There are no non-primary keys that are dependent on another non-primary key in any specific tables.
    - ➢ Hence the database is 3NF normalized.

4.4 Schema Description & Data Formats



USER	
USER_ID	CHAR(9)
PHONE_NUMBER	CHAR(10)
F_NAME	VARCHAR(20)
L_NAME	VARCHAR(20)
ITEM_CODE	CHAR(10)
ROLL_NO/EMP_ID	VARCHAR(10)
EMAIL	VARCHAR(45)
GENDER	CHAR(1)

LOST USER	
L_USER_ID	CHAR(9)

FOUND USER	
F_USER_ID	CHAR(9)

ITEM	
ITEM_ID	CHAR(10)
IMAGE	VARBINARY(MAX)
NAME	VARCHAR(20)
BRAND	VARCHAR(20)
COLOR	VARCHAR(20)
CATEGORY	VARCHAR(20)
SIZE	VARCHAR(20)

ITEM_LIST	
LIST_ITEM_CODE	CHAR(10)
NAME	VARCHAR(20)
BRAND	VARCHAR(20)
ITEM_ID	CHAR(10)
PLACE	VARCHAR(20)
TIME	DATETIME
USER_ID	CHAR(9)
COLOR	VARCHAR(20)
SIZE	VARCHAR(20)

ADMIN	
ADMIN_ID	CHAR(9)
USERNAME	VARCHAR(20)
PASSWORD	VARCHAR(25)

CONTROL	
USER_ID	CHAR(9)
ADMIN_ID	CHAR(9)

MANAGES	
ITEM_CODE_LIST	CHAR(10)
ADMIN_ID	CHAR(9)