

<PROJECT TITLE>

Database Design Document

V 3.0

By

1	<i><Group Member 1 Name></i>	<i><Roll No></i>
2	<i><Group Member 2 Name></i>	<i><Roll No></i>
3	<i><Group Member 3 Name></i>	<i><Roll No></i>



Department of Computer Sciences
Namal University
Mianwali, Pakistan

Submission Date: 30th June, 2024

REVISION HISTORY

Date	Version	Description	Approved by
<i>30/6/24</i>	<i>V.3.0</i>		
<i>10/6/24</i>	<i>V.2.0</i>		
<i>22/04/24</i>	<i>V 1.0</i>	<i>Specify the changes implemented subsequent to the submission of the previous document. These changes should be based on the suggestions given by the person who approved the document.</i>	

Instructions:

- *Place the latest revisions at the top of the table.*
- *The Revision History pertains only to changes in the document's content or any updates made after a suggestion from the approving authority. It does not apply to the template's formatting.*

TABLE OF CONTENTS

CHAPTER 1: PROJECT OVERVIEW.....	4
1.1. INTRODUCTION:	4
1.2. PROBLEM STATEMENT:.....	4
1.3. PROJECT OBJECTIVES:	4
1.4. DOCUMENT OBJECTIVES:	4
CHAPTER 2: DETAILED DATABASE DESIGN.....	5
2.1. ENTITY:	5
2.2. DATA DICTIONARY:	5
2.3. RELATIONSHIPS:	5
2.4. ENTITY RELATIONSHIP DIAGRAM:	6
CHAPTER 3 : Logical DATABASE DESIGN	7
3.1. RELATIONAL SCHEMA:	7
3.2. FUNCTIONAL DEPENDENCIES:	7
3.3. NORMALIZATION:	7
CHAPTER 4 : Physical DATABASE DESIGN.....	8
4.1. STRUCTURE OF THE TABLES:.....	8
4.2. DATA SAMPLES INSIDE TABLES:.....	8
4.3. QUERIES RESULTS:.....	8
CHAPTER 5 : Interface Design.....	9
5.1. LANGUAGE/Framework:	9
5.2. DATABASE CONNECTIVITY:.....	9
5.3. STORED PROCEDURES AND FUNCTIONS:	9
5.4. INTERFACES:.....	9
CHAPTER 6 : CONCLUSION.....	10
6.1. LESSONS LEARNED:.....	10
6.2. CHALLENGES AND SOLUTIONS:.....	10
6.3. FUTURE WORK AND IMPROVEMENTS:	10
6.4. FINAL THOUGHTS:	10
REFERENCES	11

CHAPTER 1: PROJECT OVERVIEW

1.1. INTRODUCTION:

In this section, you should provide an overview of your database project. The introduction is the foundation for understanding the purpose, scope, and significance of your database design and implementation.

1.2. PROBLEM STATEMENT:

In the "Problem Statement" section, you will outline the precise issues or challenges your database project intends to tackle. This section acts as a central point for recognizing the shortcomings or inadequacies of the current system or procedure, thereby driving the necessity for devising a new database solution. Conclude the problem statement by describing the desired future state. Explain how implementing your database solution will address the identified issues and enhance the processes' efficiency, effectiveness, or quality.

1.3. PROJECT OBJECTIVES:

This section presents the specific goals and outcomes your database project aims to achieve. Use bullet points to list each objective clearly and succinctly. Clearly define each objective in terms of what you intend to accomplish. Avoid vague language and ensure that objectives are specific, measurable, achievable, relevant, and time-bound (SMART).

1.4. DOCUMENT OBJECTIVES:

In this section, you will outline the specific objectives of this project documentation. These objectives help guide the content and structure of your documentation, ensuring that it effectively communicates the essential information about your database project. Outline the structure and organization of this documentation.

CHAPTER 2: DETAILED DATABASE DESIGN

2.1. ENTITY:

In the "Entities" section, identify and define the entities that are central to the database project. Begin by identifying the entities that are relevant to your database project. These could be objects, concepts, or events that you need to keep track of in your database system. For each identified entity, provide a clear and concise definition. Describe what the entity represents in the context of your database project and its role within the system.

Sr. No	Entity Name	Description
01	User	A user is an individuals who have registered or created accounts within the system.

2.2. DATA DICTIONARY:

In the "Attributes" section, identify and define the specific characteristics or properties of each entity in the database. Provide a list of attributes associated with the chosen entity. Each attribute should represent a specific piece of information or characteristic that has to be stored in the database. Specify the data type for each attribute to indicate the kind of data it will store. Common data types include text, numbers, dates, Boolean values. Identify any constraints or rules that apply to the attribute values. This may include constraints such as required fields, unique values, minimum or maximum lengths, or format requirements. For each attribute, provide a clear and concise definition or description. Explain what the attribute represents and its significance within the context of the entity.

2.2.1. User:

Sr. No	Name	Data Type	Constraint	Description
01	Email ID (Primary Key)	String	NOT NULL	Email ID of the user
02	Attribute 2	Data type	Any constraint	Definition of Attribute

2.3. RELATIONSHIPS:

In the "Relationships" section, define the connections or associations between entities in the database. Begin by specifying the entities involved in the relationship. Clearly state the names of the entities to provide context for the relationship that will be defined. Identify the type of relationship between the entities. Describe the nature of the relationship and the cardinality (e.g., each entity instance may have one or multiple related instances).

Sr. No	Participating Entities	Relation	Business Rule
<i>01</i>	<i>User , File</i>	<i>User saves File</i>	<i>A User may save more than one file. A file is saved by exactly one user.</i>

2.4. ENTITY RELATIONSHIP DIAGRAM:

Draw the ERD Diagram of your project. List all the entities, attributes, relationships, and primary keys of the entities and show the cardinalities between the entities. List any assumptions that you made during making the ERD. Select a tool for creating your ERD. You can use specialized software such as Microsoft Visio, Lucidchart, or online ERD tools. Ensure all the text, relationships, and cardinalities are readable in the picture.

CHAPTER 3 : LOGICAL DATABASE DESIGN

3.1. RELATIONAL SCHEMA:

In this section, document the process of converting your Entity-Relationship Diagram (ERD) into a relational schema. Begin by listing each relation, specifying primary keys, and detailing the attributes as columns. Clearly outline how relationships are represented through foreign keys and bridge/associative relations for many-to-many relationships. This section should concisely capture the transformation from your conceptual design to an implementable database structure, regardless of the choice of DBMS.

3.2. FUNCTIONAL DEPENDENCIES:

For each table in your relational schema, list the functional dependencies. Use the format " $A \rightarrow B$ " to indicate that attribute A functionally determines attribute B. Ensure you cover all relevant functional dependencies for each table. Provide examples to illustrate the functional dependencies. Use sample data or scenarios to make the dependencies more concrete and understandable.

3.3. NORMALIZATION:

Using the identified functional dependencies, provide partial and transitive functional dependencies. Provide an overview of the steps you took to normalize your tables, mentioning specific normal forms (e.g., 1NF, 2NF, 3NF) achieved and any transformations made to satisfy the rules of each normal form. Highlight key decisions and changes made to address functional dependencies and eliminate anomalies. This section must contain all 3NF relations of the defined system at the end.

CHAPTER 4 : PHYSICAL DATABASE DESIGN

4.1. STRUCTURE OF THE TABLES:

In this section, you are required to document the structure of each table in your database. Insert the SQL queries used to show the structure of each table and include the output that shows the structure. This output should detail the columns, data types, and any constraints applied to each column, such as primary keys, foreign keys, unique constraints, and default values. Ensure that the structure of all tables is clearly presented, providing a comprehensive view of your database schema.

4.2. DATA SAMPLES INSIDE TABLES:

Here, you must provide sample data for each table in your database. Insert the SQL queries that retrieve all data from each table and include the output of these queries. This output should display the actual data rows in each table, demonstrating that your database is populated with sample data. Make sure to show sufficient rows to represent the data realistically, with at least 15 rows per table.

4.3. QUERIES RESULTS:

In this section, you must write at least 10 different SQL queries that retrieve various results from your database. For each query, provide the SQL statement along with the output it generates. These queries should demonstrate a range of functionalities and complexities, including but not limited to:

- *Queries involving joins (including those with more than two tables).*
- *Aggregate queries using GROUP BY and HAVING clauses.*
- *Queries with ORDER BY clauses.*
- *Nested subqueries.*

The outputs of these queries should clearly show the results returned by the database, illustrating your ability to manipulate and retrieve data effectively using SQL.

CHAPTER 5 : INTERFACE DESIGN

5.1. LANGUAGE/Framework:

In this section, detail the chosen technology, programming language, or framework used for developing the GUI client. Explain why this particular technology was selected for the project, highlighting its suitability and advantages. Discuss the key features of the chosen technology and how they contribute to the development and functionality of the GUI client.

5.2. DATABASE CONNECTIVITY:

Explain how the GUI client connects to the MySQL database, including the specific libraries or frameworks used to facilitate this connection. Provide code snippets that demonstrate how the database connection is set up, ensuring to include any necessary error handling procedures.

5.3. STORED PROCEDURES AND FUNCTIONS:

Provide the scripts for the stored procedures and functions used in the database. For each procedure and function, explain its objective or purpose within the application. Describe what each script does, how it is implemented, and how it enhances the functionality of the GUI client.

5.4. INTERFACES:

Provide visual documentation of the GUI client by including screenshots of the main interface, forms for data entry, data display elements, and other key functionalities. Each screenshot should be accompanied by a brief description explaining the functionality shown.

CHAPTER 6 : CONCLUSION

6.1. LESSONS LEARNED:

Summarize the important lessons learned during the project. This can include technical insights, project management skills, teamwork experiences, and any other valuable takeaways.

6.2. CHALLENGES AND SOLUTIONS:

Reflect on the major challenges faced throughout the project and how they were addressed. Discuss any obstacles encountered during the design, implementation, and testing phases, and explain the solutions or strategies used to overcome them.

6.3. FUTURE WORK AND IMPROVEMENTS:

Suggest potential future enhancements and areas for improvement. Discuss how the project can be expanded or refined, including additional features, optimizations, or broader applications of the developed system.

6.4. FINAL THOUGHTS:

Conclude with any final remarks or reflections on the project. This can include personal insights, the overall impact of the project, and any concluding comments or acknowledgments.

REFERENCES

Provide a list of all the sources cited or consulted during the development of the database project. This section serves to acknowledge the contributions of other authors and researchers, as well as to enable readers to locate the sources for further information. Ensure that references are formatted according to IEEE referencing style. Also, provide the in-text citations of the references

Notes to the Author

[This document is a template for creating a Database Design Document (DDD) for a given project. The template includes instructions to the author, boilerplate text, and fields that should be replaced with the values specific to the particular project.]

- *Blue italicized text enclosed in square brackets (i.e., [text]) provides instructions to the document author; or describes the intent, assumptions and context for content included in this document.*
- *Blue italicized text enclosed in angle brackets (i.e., <text>) indicates a field that should be replaced with information specific to the particular project.*
- *Text and tables in black are provided as boilerplate examples of wording and formats that may be used or modified as appropriate.*

Delete this “Notes to the Author” page and all instructions to the author (i.e., all blue italicized text enclosed in square brackets) before finalizing the initial draft of the DDD.]

Formatting Guidelines:

- *Level 1 Heading: Font Style: Times New Roman, Font Size: 20, Color: Black, Case: All Caps, Align: Right, Numbering Style: CHAPTER 1... Should appear in Table of Content*
- *Level 2 Heading: Font Style: Times New Roman, Font Size 16, Color Black, Case: All Caps, Align: Left, Numbering Style: 1.1, 1.2 ... Should appear in Table of Content*
- *Level 3 Heading: Font Style: Times New Roman, Font Size 14, Color Black, Case: Capitalize each word, Align: Left, Numbering Style: 1.1.1, 1.1.2 ... Should Not appear in Table of Content*
- *Paragraph: Font Style: Times New Roman, Font Size 12, Color Black, Case: Sentence Case, Align: Justified, Should Not appear in Table of Content, Add space before and after paragraph.*
- *Insert caption to the picture with figure number.*
- *Insert caption to table with table number.*

Any text that is in blue and italicized is for your understanding only and should not be included in your submission document.