

Institution Name

Course / Unit

Title of the Work

Student Name(s)

Supervisor: Supervisor / Instructor

Month Year

Abstract

This report addresses *[context/problem]* by proposing *[solution/system]*. The objectives are *[objective 1]*, *[objective 2]*, and *[objective 3]*. The proposed solution is based on *[architecture/approach]* and a database model designed to enforce integrity through *[constraints/normalization]*. Evaluation was performed using *[testing approach]* and *[performance/validation method]*. Results show *[main result]*, while the main limitations are *[limitations]*.

Keywords: software engineering, databases, data modelling, constraints, testing

Contents

1	Introduction	6
1.1	Context and Motivation	6
1.2	Problem Statement	6
1.3	Objectives	6
1.4	Scope and Limitations	6
1.5	Document Structure	7
2	Background and Related Work	8
2.1	Background Concepts	8
2.2	Related Work	8
2.3	Positioning	8
3	Requirements Analysis	9
3.1	Users and Stakeholders	9
3.2	Functional Requirements	9
3.3	Non-Functional Requirements	9
3.4	Use Cases / User Stories	10
3.5	Acceptance Criteria (Recommended)	10
4	System Design	11
4.1	Architecture Overview	11
4.2	Component Responsibilities	11
4.3	Key Workflows (Optional)	11
4.4	Design Decisions	11
5	Database Design	12

5.1	Conceptual Model (ER Diagram)	12
5.2	Logical Model (Relational Schema)	12
5.3	Normalization and Rationale	12
5.4	Integrity Constraints	13
5.5	Indexes and Query Rationale	13
6	Implementation Overview	14
6.1	Technology Stack	14
6.2	Traceability (Requirements to Implementation)	14
6.3	Security and Validation	14
7	Testing and Evaluation	15
7.1	Testing Strategy	15
7.2	Test Results	15
7.3	Performance Evaluation (Basic but Real)	15
8	Results and Discussion	16
8.1	Results	16
8.2	Limitations	16
8.3	Trade-offs	16
9	Conclusion and Future Work	17
9.1	Conclusion	17
9.2	Future Work	17
A	Appendix A: SQL Schema (excerpt)	18
B	Appendix B: Representative Queries	19
C	Appendix C: Extra Evidence	20

List of Figures

4.1	High-level architecture.	11
5.1	Entity-Relationship diagram.	12

List of Tables

3.1	Use Cases	10
5.1	Schema Summary (template)	12
6.1	Traceability Table (template)	14
7.1	Testing Summary (template)	15

Chapter 1

Introduction

1.1 Context and Motivation

Explain the real context. Describe why the problem exists and why it matters. Give one or two concrete examples of failures or inefficiencies that motivate a solution.

1.2 Problem Statement

State the problem clearly in 3–6 sentences. Avoid broad statements such as “there is a need for better systems”. Instead, define what is missing and what must be guaranteed.

1.3 Objectives

List 3–6 objectives. Each objective should be measurable.

- **O1:** *[objective]*
- **O2:** *[objective]*
- **O3:** *[objective]*

1.4 Scope and Limitations

Clarify what is included and excluded. Mention scale assumptions (number of users, records, etc.) if relevant.

1.5 Document Structure

Explain briefly what each chapter contains.

Chapter 2

Background and Related Work

2.1 Background Concepts

Summarize the concepts needed to understand your work. For database projects, typical concepts include ER modelling, normalization, constraints, indexes, and transactions. Cite sources for definitions and claims.

2.2 Related Work

Describe similar systems or approaches (academic or practical). Explain what you reuse and what is different in your work. Use 2–10 references depending on the assignment scope.

2.3 Positioning

One paragraph explaining the focus of this work (e.g., correctness and integrity vs feature richness).

Chapter 3

Requirements Analysis

3.1 Users and Stakeholders

Who will use the system? In what situations? What are their goals?

3.2 Functional Requirements

Write requirements as “The system shall . . .” statements. Number them to enable traceability later.

FR1 The system shall *[do something]*.

FR2 The system shall *[do something]*.

FR3 The system shall *[do something]*.

3.3 Non-Functional Requirements

Include at least 4 categories: integrity, performance, security, maintainability/usability.

NFR1 **Integrity:** *[constraints/invariants that must hold]*.

NFR2 **Performance:** *[response time for key operations, dataset scale]*.

NFR3 **Security:** *[who can do what; protection against common risks]*.

NFR4 **Maintainability:** *[modularity, tests, documentation]*.

3.4 Use Cases / User Stories

List 4–8 key use cases. Describe the main success path and the main failure path.

Table 3.1: Use Cases

UC ID	Description
UC1	<i>[short description of workflow]</i>
UC2	<i>[short description of workflow]</i>
UC3	<i>[short description of workflow]</i>

3.5 Acceptance Criteria (Recommended)

Write 1–2 acceptance criteria for each key requirement (what must be true to consider it done).

Chapter 4

System Design

4.1 Architecture Overview

Describe the architecture at a high level. Include a diagram (layered architecture is acceptable for bachelor projects).

Placeholder: Architecture diagram (components and interactions)

Figure 4.1: High-level architecture.

4.2 Component Responsibilities

Describe each component/module and what it owns. Keep it short and precise.

4.3 Key Workflows (Optional)

For 1–2 critical workflows, provide a sequence diagram or a structured description.

4.4 Design Decisions

Explain 3–6 important decisions and trade-offs. Examples: relational vs NoSQL, normalization level, transaction approach, indexing strategy.

Chapter 5

Database Design

5.1 Conceptual Model (ER Diagram)

Provide an ER diagram with cardinalities. Explain the main entities and relationships.

Placeholder: ER diagram with entities and cardinalities

Figure 5.1: Entity-Relationship diagram.

5.2 Logical Model (Relational Schema)

Describe tables, attributes, primary keys, foreign keys, and constraints. Add a short schema summary table.

Table 5.1: Schema Summary (template)

Table	Notes (PK/FK/Constraints)
<i>table_1</i>	PK(<i>id</i>), FK(...), UNIQUE(...), CHECK(...)
<i>table_2</i>	PK(<i>id</i>), FK(...), NOT NULL(...)
<i>table_3</i>	PK(<i>id</i>), FK(...), CHECK(...)

5.3 Normalization and Rationale

State the target normal form (e.g., 3NF). Justify the choice in 1–2 paragraphs. Mention any denormalization and why it is acceptable.

5.4 Integrity Constraints

List the key invariants and how they are enforced: PK/FK, uniqueness, domain checks, and referential actions (ON DELETE/ON UPDATE).

5.5 Indexes and Query Rationale

Provide 3–6 critical queries (in text) and explain which indexes support them. Optionally include EXPLAIN summaries.

Chapter 6

Implementation Overview

6.1 Technology Stack

List the tools and technologies used and provide short justifications.

6.2 Traceability (Requirements to Implementation)

Provide a table connecting requirements to modules/endpoints/queries/tests.

Table 6.1: Traceability Table (template)

Req	Implementation	Evidence
FR1	<i>[module / endpoint / workflow]</i>	<i>[test / screenshot]</i>
FR2	<i>[module / endpoint / workflow]</i>	<i>[test / screenshot]</i>
NFR1	<i>[constraints / security control]</i>	<i>[schema / test]</i>

6.3 Security and Validation

Summarize authentication/authorization and input validation. Mention parameterized queries and error handling conventions.

Chapter 7

Testing and Evaluation

7.1 Testing Strategy

Describe the levels of testing (unit, integration, system). Specify what is tested at each level.

7.2 Test Results

Summarize tests in a table and mention the most important edge cases.

Table 7.1: Testing Summary (template)

Area	What was tested	Result
Business rules	<i>[edge cases and invalid states]</i>	<i>Pass/Fail</i>
Database integrity	<i>[FK, CHECK, UNIQUE]</i>	<i>Pass/Fail</i>
Critical workflows	<i>[UC1..UCk]</i>	<i>Pass/Fail</i>

7.3 Performance Evaluation (Basic but Real)

State dataset size, test method, and results. Include at least one critical query and show the impact of indexes if possible.

Chapter 8

Results and Discussion

8.1 Results

Summarize what was achieved, mapped to objectives and requirements.

8.2 Limitations

List honest limitations: missing features, scale limits, incomplete security, etc.

8.3 Trade-offs

Explain important trade-offs (correctness vs complexity, normalization vs query speed, etc.).

Chapter 9

Conclusion and Future Work

9.1 Conclusion

One short section summarizing the outcome and main evidence (tests, constraints, evaluation).

9.2 Future Work

List realistic improvements.

- *[improvement 1]*
- *[improvement 2]*
- *[improvement 3]*

Appendix A

Appendix A: SQL Schema (excerpt)

-- Put an excerpt of your schema here, not the whole thing if it is very long.
-- Include CREATE TABLE + key constraints + key indexes.

Appendix B

Appendix B: Representative Queries

-- Include 3--10 queries that demonstrate your system.

-- Show at least: one join query, one aggregation query, and one integrity-related query

Appendix C

Appendix C: Extra Evidence