(USED THE LATE GRACE HOURS-72 SO SUBMITTED THE ASSIGNMENT LATE)

CMPSC 431W Project – Stage III Report

Kaushik Bhargav, Apoorv Singh

12/08/2024

1. Project Overview

The goal of our project was to design and implement a database system to manage an organization's employees, departments, projects, tasks, schedules, resources, and other related entities. The system was designed to capture and manage data related to employee assignments, project progress, scheduling, and resource allocation.

The project was divided into three stages and each stage was completed successfully, with specific tasks accomplished:

Stage I:

- Identified key system requirements for managing employees, departments, projects, tasks, schedules, resources, and skills1
- Defined key entities and their attributes, as well as relationships between them1
- Planned interface requirements and made crucial decisions for the project foundation

.

Stage II:

- Developed an ER model with 9 entities and 12 relationships.
- Ensured all relationships were clearly defined with cardinalities and foreign key mappings
- Refined the schema to comply with BCNF and normalized all tables
- Wrote SQL DDL statements to create the database schema
- Developed SQL queries to support proposed functionalities

Stage III:

- Implemented the database schema in MySQL
- Created backend files to build a complete system
- Used Django to execute database management queries
- Developed multiple pages for user navigation and application use
- Created a user manual for system deployment and usage
- Hosted final source code and SQL scripts in a Git repository

This report reflects on the work completed during the semester, the development process, and how it aligns with the team contract.

2. Project Work Summary

2.1 Work Carried Out During the Semester

Stage I: Requirements Gathering and Conceptual Design

In Stage I, the project team focused on identifying the key requirements necessary for the system, which aimed to manage various organizational components such as employees, departments, projects, tasks, schedules, resources, and skills. The team defined essential entities and their attributes, establishing relationships between them to ensure a well-structured database design. This foundational work included planning interface requirements, which were crucial for creating a user-friendly experience. The decisions made during this stage laid the groundwork for subsequent stages of development.

Stage II: ER Modeling and Schema Design

Stage II involved the development of an Entity-Relationship (ER) model that encompassed 9 entities and 12 relationships. The team ensured that all relationships were clearly defined with appropriate cardinalities and foreign key mappings. The schema was meticulously refined to comply with Boyce-Codd Normal Form (BCNF), guaranteeing that all tables were normalized to eliminate redundancy and maintain data integrity. SQL Data Definition Language (DDL) statements were crafted to create the database schema, while SQL queries were developed to support the functionalities proposed in the requirements gathering phase. This stage was critical in transforming conceptual designs into a structured database format ready for implementation.

Stage III: Final Implementation and Testing

In Stage III, the team implemented the database schema within a relational database management system (MySQL). They created the database along with all necessary backend files to ensure a fully functional application. The integration of Django facilitated the execution of database management queries, allowing for seamless interaction between the application and the database. To enhance user experience, multiple pages were developed for easy navigation and application use. Additionally, a comprehensive user manual was created to guide users on deploying and utilizing the system effectively. Finally, all source code and SQL scripts were hosted in a Git repository for submission, marking the completion of the project.

This structured approach across all stages ensured that the project was executed efficiently and successfully met its objectives.

2.2 Comparison with the Team Contract

According to our team contract, we agreed to: Divide the work equally between team members, Meet weekly to discuss progress and resolve any issue and Complete each stage of the project by the assigned deadlines.

Reflection:

Work Distribution:

According to the team contract, work was to be divided equally between team members. This was successfully implemented throughout the project:

- Kaushik focused on schema design and SQL implementation
- Apoorv concentrated on testing, front-end and documentation

Both members applied their identified skill strengths effectively:

- Apoorv utilized work management skills for designing front-end, outlining responsibilities and report preparation
- Kaushik applied teamwork skills and backend/SQL expertise for developing the website/app and connecting it to the database

Meetings:

The contract stipulated:

- Maintaining a team chat group for quick updates
- Scheduling regular meetings for in-depth discussions

In practice:

- Weekly meetings as planned couldn't be held consistently
- However, online collaboration tools were used effectively to stay on track and maintain communication

Deadlines:

The team contract emphasized meeting all project milestones as outlined in the course schedule. The project execution largely aligned with this:

- All stages were completed on time, adhering to the project timeline
- There was a slight delay in testing the final system, but overall deadlines were met with the extra grace hours

3. Reflection on the Project Development Process

3.1 Challenges Faced

Initial Scope and Requirements Definition

In the early stages of the project, the team encountered difficulties in fully grasping the project's scope and defining comprehensive requirements. While they had a general understanding, pinpointing specific elements that would lead to desired outcomes proved challenging. This obstacle was overcome through:

- Collaborative discussions between team members
- Iterative refinement of project goals and requirements
- Careful analysis of the organization's needs for managing employees, departments, projects, tasks, schedules, and resources

Schema Design and BCNF Implementation

A significant challenge arose in fully comprehending and implementing Boyce-Codd Normal Form (BCNF):

- The team had to carefully analyze functional dependencies for each table
- Ensuring all tables complied with BCNF required meticulous review and refinement
- The process involved checking that for every non-trivial functional dependency X → Y, X was a superkey
- This challenge was particularly evident in the design of complex tables like Employee, Project, and Task, which had multiple attributes and relationships.

Testing and Integration

The testing phase presented several hurdles:

- Extensive troubleshooting was required, particularly in modifying table queries
- Linking the database to the webpage proved to be one of the most problematic aspects
- The team faced issues in ensuring seamless data flow between the frontend and backend, which was crucial for the application's functionality

3.2 Lessons Learned

Effective Collaboration

The project underscored the importance of:

- Clear and consistent communication among team members
- Efficient task delegation based on individual strengths (e.g., Kaushik focusing on backend and SQL, Apoorv on testing and documentation)
- Adapting to challenges in meeting schedules by utilizing online collaboration tools

Advanced Database Design Skills

- Through this project, the team significantly enhanced their understanding of:
- Database normalization techniques, particularly BCNF
- ER modeling, including the creation of a complex model with 9 entities and 12 relationships
- Schema refinement processes to ensure data integrity and eliminate redundancy
- Frontend development and its integration with backend database systems

SQL Proficiency

- The project provided extensive hands-on experience in:
- Writing and optimizing complex SQL queries for various functionalities
- Developing DDL statements for creating a robust database schema
- Implementing and testing queries for diverse operations like employee management, project tracking, and resource allocation

3.3 Improvements for Future Projects

Time Management

- Allocate additional time for testing and debugging, particularly for integration between frontend and backend systems
- Plan for potential delays in final system testing to avoid last-minute rushes

Project Management Tools

- Implement more sophisticated project management tools to track progress effectively
- Use tools that can help in visualizing project timelines and task dependencies

Requirement Analysis

- Conduct a more thorough and structured requirement analysis in the initial stages
- Involve stakeholders early in the process to ensure all perspectives are considered
- Develop more detailed use cases to guide the development process

Version Control and Documentation

- Utilize version control systems like Git more effectively from the project's inception
- Maintain comprehensive documentation throughout the development process, not just in the final stages

4. Git Repository

The final source code and SQL scripts are hosted in the following Git repository:

Git Repository URL: https://github.com/SKB-18/Task-Scheduler