Project and Work Tracking System

Project documentation

Mentor: Prof. Klen Čopič Pucihar, PhD

Authors: Una Novaković and Luka Matič

University of Primorska

FAMNIT

2026

Contents

1. Problem description

Modern organizations often manage multiple projects simultaneously, involving different employees, tasks, and deadlines. In smaller companies or teams, project coordination and task tracking are frequently handled using informal tools such as spreadsheets, email communication, or note-taking applications. While these methods may be sufficient for smaller workloads, they quickly become inefficient as the number of projects and tasks increases.

Without a centralized system, project managers may struggle to maintain an organized overview of ongoing work. Information about tasks, responsibilities, and deadlines can become scattered across different communication channels, making it more difficult to monitor project progress or identify potential delays. Employees may also experience difficulties when recording work hours or identifying their assigned responsibilities, which can lead to inconsistencies in reporting and reduced transparency within the team.

To address these challenges, organizations benefit from using an information system that centralizes project management activities. Such a system provides a structured way to organize projects, assign tasks to employees, track deadlines, and record working hours. By storing all relevant information within a single platform, both managers and employees gain improved visibility into project progress, responsibilities, and overall team activity.

2. System analysis and design

2.1 Functional requirements

1. User authentication.

   The system shall provide user authentication functionality that allows registered users to securely log into the system using their credentials.

2. Project overview and management**.**

   The system shall provide an overview of projects assigned to users, including relevant project information such as project name, description, start date, end date, project status,

progress overview. Project managers and administrators shall be able to create and manage projects.

3. Task management.

   The system shall provide task management functionality. Project managers and administrators shall be able to: create tasks, assign tasks to employees, define deadlines, monitor task progress. Employees shall be able to: view assigned tasks, update task status, add descriptions of completed work, mark tasks as completed.

4. Work hour recording.

   The system should allow employees to record the number of hours worked on assigned tasks. Employees shall be able to select a task, enter work date, record hours worked, add a work description.

5. Project progress reporting.

   The system shall provide reporting functionality that summarizes project progress. The system shall display: completed tasks, total tasks, logged work hours, project completion percentage. This functionality allows project managers to monitor the overall status and progress of projects.

2.2 Non-functional requirements

Performance

The system should provide response times below 2 seconds for common operations such as user login, project management, task updates, and time entry recording under normal usage conditions.

Availability

The system should be available whenever the local server environment is running. Maintenance activities should not require extensive downtime.

Security

Users must authenticate using their email address and password before accessing protected functionality. Passwords are stored using secure hashing mechanisms. Sensitive data should be transmitted over HTTPS when deployed to a production environment.

Usability

New users should be able to navigate the system and perform common tasks such as creating projects, updating tasks, and recording work hours with minimal training. The user interface should remain clear and intuitive.

Reliability

The system should ensure that project, task, user, and time entry data are stored consistently in the database and are not lost during normal operation.

Scalability

The system should support the addition of new users, projects, tasks, and time entries without requiring major architectural changes.

Maintainability

The code should be organized into separate frontend, backend, and database components to simplify future maintenance and further development.

Compatibility

The system should function correctly in modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.

2.3 User roles

Administrator

- Manage users
- Manage projects
- Manage tasks
- View project progress
- Record work hours

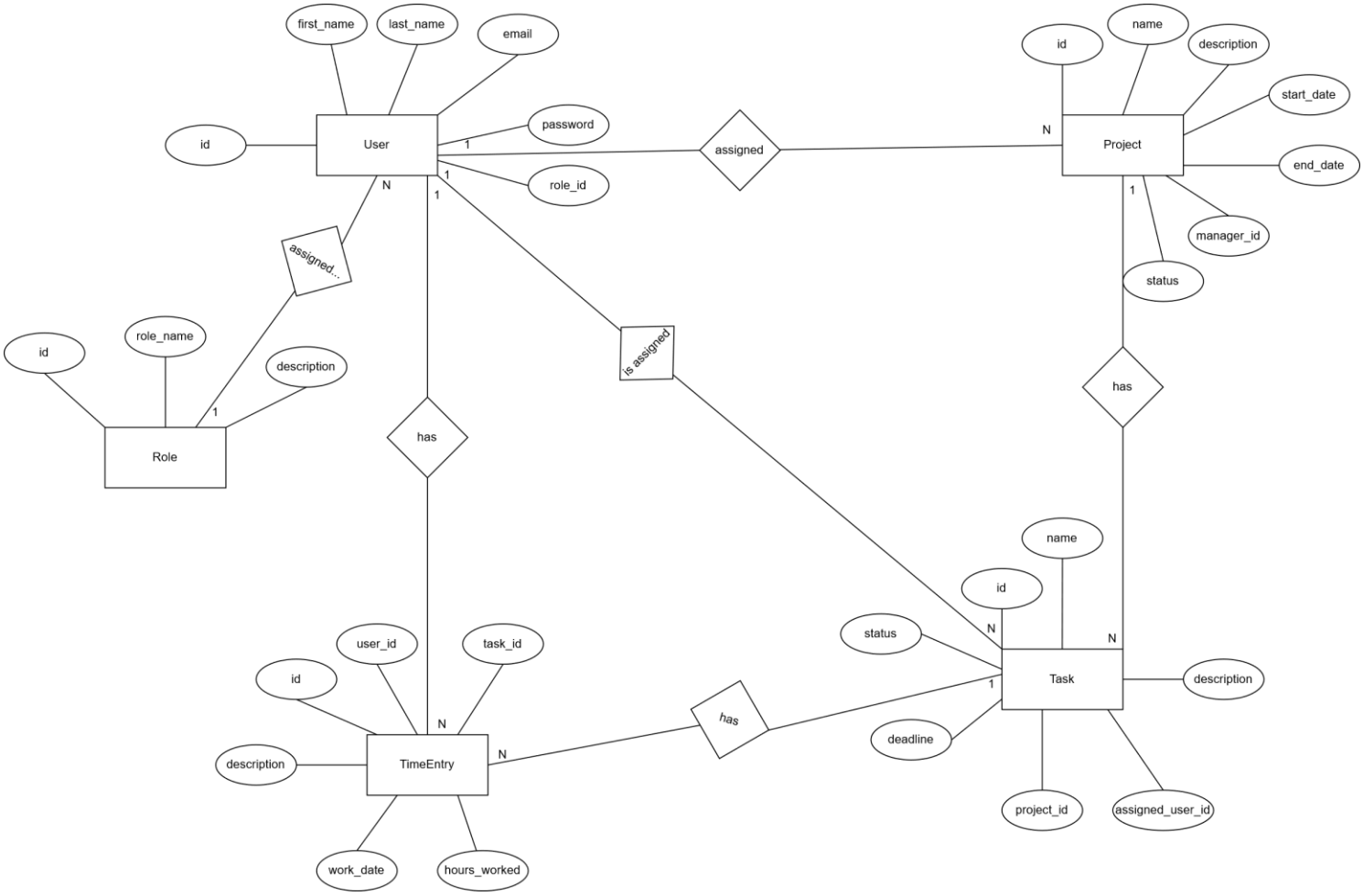
Project Manager

- Create and manage projects
- Create and assign tasks
- View project progress
- Record work hours

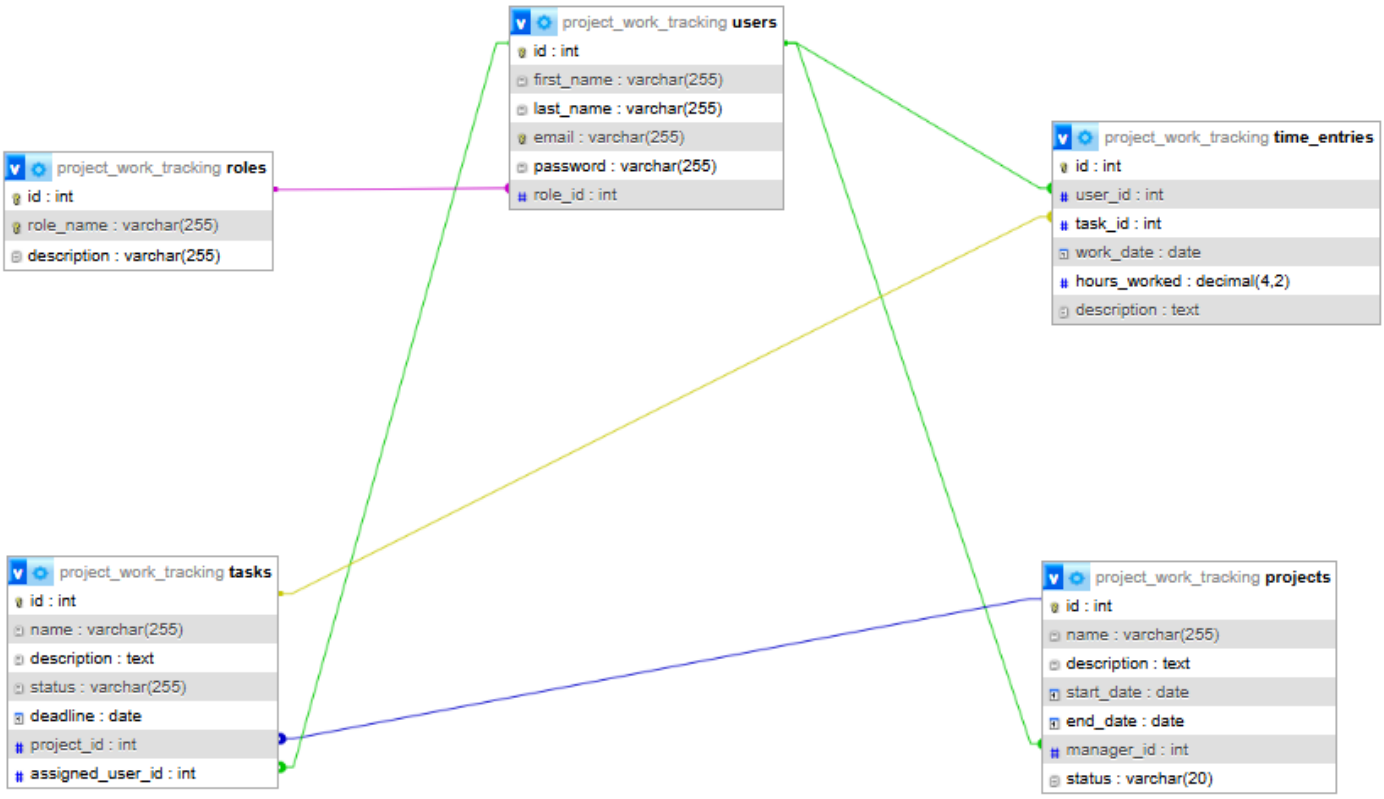
Employee

- View assigned tasks
- Update task status
- Record work hours
- View project progress

2.4 ER diagram



2.5 Relational model



2.6 Database description

The system uses a MySQL relational database consisting of five tables:

users – stores user accounts and role information
roles – stores available user roles
projects – stores project information
tasks – stores tasks associated with projects
time_entries – stores recorded work hours

Relationships between the tables ensure data consistency and support project management, task assignment, and work-hour tracking.

2.7 Technology stack

Layer	Technology
Frontend	**HTML, CSS, Bootstrap**
Backend	**PHP**
Database	**MySQL**
Development Environment	**XAMPP**
Version Control	**Git & GitHub**

The system was developed using PHP for backend functionality and MySQL for data storage. Bootstrap was used to create a responsive and user-friendly interface. Development and testing were performed locally using XAMPP, while Git and GitHub were used for version control and collaboration.

3. System architecture

3.1 System layers

Presentation Layer

- User interface
- Forms and pages
- Bootstrap styling

Business Logic Layer

- Authentication
- Permission checks
- Project, task and time entry processing

Data Layer

- MySQL database
- Storage of users, projects, tasks and time entries

3.2 Modules

Authentication Module
User registration, user login, session management, password hashing.

Project Management Module
Creating projects, editing projects, assigning project managers, viewing project information.

Task Management Module
Creating tasks, assigning tasks to employees, setting deadlines, updating task status, recording completed work.

Time Tracking Module
Logging work hours, associating hours with tasks, viewing and editing time entries.

Project Progress Module

Progress is calculated as:

$$Progress = \frac{Completed\ Tasks}{Total\ Tasks} \times 100$$

Total logged hours are displayed for each project. Managers can monitor project completion status.

3.3 Development process

The project was developed using an iterative approach inspired by Scrum practices. Work was divided into smaller features, which were implemented independently and later integrated into the main branch.

3.4 Version Control

Git and GitHub were used for version control. Each feature was developed in a separate branch and merged through pull requests after testing and review.
Source code repository: https://github.com/unvkvc/project-work-tracking-system