

Curiem: Customer relationship management

CRM web-based application (Spring Boot + React)

Anshika Aggarwal, Raj Kumar

Department of Computer Science Engineering
KCC Institute of Technology and Management, Greater Noida, India
anshikaggarwalcs@gmail.com, rajkumar.rk0104@gmail.com

Abstract

Customer relationship management (CRM) systems play an important role in managing customer information, improving communication, and supporting business decision-making. Due to the rising trend of using web technology, there is a growing need for a CRM tool that is highly flexible and easy to use. In this paper, the design and implementation of a web-based CRM system, named CURIEM, is discussed, where the backend is developed using Spring Boot and the frontend is developed using React. In the proposed system, there are functionalities like customer data management, interactions, and a role-based access system. Functional evaluation shows that the system can be effectively employed for performing the basic operations of a CRM system and can also be employed by small and medium-scale businesses.

I. INTRODUCTION

Customer Relationship Management (CRM) systems are widely used by organizations to manage customer information, monitor interactions, and support marketing, sales, and service processes. With the rapid growth of web technologies and digital business platforms, organizations increasingly rely on web-based CRM solutions to improve efficiency, accessibility, and customer engagement.

This paper presents CURIEM, which is a web-based customer relationship management system integrating customer data management with project and task tracking within a single platform. The system uses Spring Boot for the implementation of the backend services and React for the front-end interface. It leverages such capabilities like role-based access control, dashboard-based analytics, or automatic tracking of tasks to provide full transparency and coordination in workflows.

The objective of this work is to design and implement a practical CRM solution that overcomes some of the known limitations of existing tools by providing an integrated, secure, easy-to-use platform suitable for small and medium-scale organizations. The rest of the paper presents the identified research gaps, methodology of the system, architecture and implementation details, followed by evaluation results and conclusions.

II. PROBLEM STATEMENT AND RESEARCH GAP

Many existing Customer Relationship Management (CRM) systems primarily focus on sales pipelines or customer support operations. While these functionalities are essential, they often lack seamless integration with internal project execution and task management. As a result, organizations are required to use multiple independent tools to manage customers, projects, and team workflows.

This fragmented approach leads to several challenges, including duplication of data, inconsistent reporting, reduced transparency, and inefficient coordination among teams. These issues are particularly significant for small and medium-scale enterprises (SMEs), which typically operate with limited resources and require centralized systems to streamline operations.

Another limitation observed in existing CRM platforms is the lack of flexible role-based access control and real-time analytics aligned with organizational workflows. Managers often lack a unified view of project progress and task status, while users may have access to unnecessary system modules.

Research Gap:

There is a need for an integrated web-based CRM solution that combines customer management, project tracking, task monitoring, role-based access control, and analytics within a single platform.

Proposed Solution:

To address this gap, this paper proposes **CURIEM**, a web-based CRM system developed using Spring Boot and React, designed to support efficient workflow coordination and operational transparency for small and medium-scale organizations.

III. RESEARCH METHODOLOGY

The paper adopts an applied research methodology approach for designing, developing, and evaluating a practical software system. The purpose of the proposed research study is to develop a functional Customer Relationship Management (CRM) software system to address the shortcomings present in the traditional CRM software systems and project managing software systems used by professionals.

The proposed system development process will follow the "Agile Software Development Life Cycle (SDLC)" because of its iterative nature. In the Agile method, system functionalities will be continuously improved after testing and other functional requirements.

3.1 Requirement Analysis

The requirements were arrived at after analyzing the existing CRM solutions and project management software, and noting the challenges of workflow processes that small and medium-sized businesses face. There were functional and non-functional requirements of the system. The requirements included customer management, project management, role-based security, analysis and statistics, management of events, and authentication. The requirements included security, scalability, usability, and performance.

3.2 System Design

The system architecture has been modeled in such a way that it has adopted a modular design for easier scalability and maintainability. The front end of the system has adopted React to make it dynamic, while the back end services have adopted Spring Boot to provide RESTful services to carry out business operations on data. The system has adopted MySQL to model its relational database with entities like users, clients, projects, and tasks.

3.3 Implementation

The final stage involved building backend services with Spring Boot, Spring Security, and JWT authentication for secure role-based handling. REST APIs were built to allow CRUD operation handling in various modules at the backend. Front-end programming was done with React and Redux. Other technologies incorporated included AWS S3.

3.4 Testing and Evaluation

System testing involved various levels, such as unit testing of backend modules, integration testing of RESTful API using Postman, and functional testing of the user interface. Role-based access functionalities were also validated to check that only authorized modules can be accessed by the users. Testing involved various aspects of the system, including its functionalities, usability, security, and performance at different levels of roles.

IV. SYSTEM ARCHITECTURE

The architecture pattern adopted in the CURIEM system is based on a three-tier web application architecture. Therefore, this is done with the aim of creating a sustainable, secured, and easily maintained system. It is made up of three distinct tiers, namely the presentation tier(frontend), application tier(backend), and data storage tier. The overall system architecture is illustrated in Fig. 1.

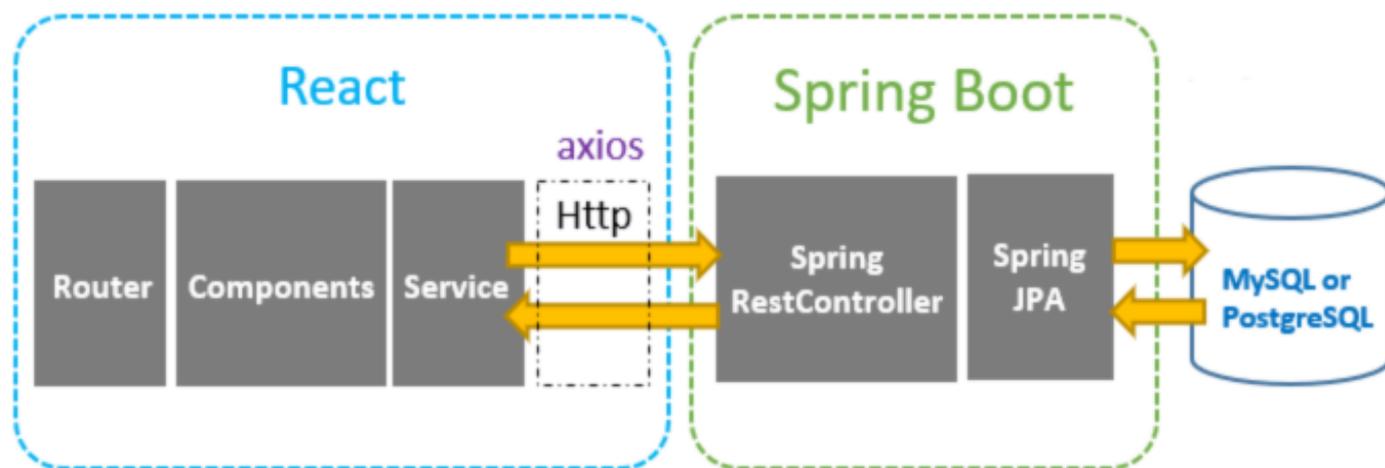


Fig. 1. Three-tier architecture of the CURIEM CRM web application

- Frontend Technology

The frontend layer is built with React, which gives it a dynamic and responsive UI. The UI interacts with the user, performs data visualization, and allows role-based navigation. The React components in the UI communicate with the backend components through RESTful API, thus achieving separation of concerns between the UI and business logic. These technologies include JavaScript, React, Redux.

- Backend Technology Used

The backend layer is developed using Spring Boot, which takes care of all the logic, authentication, authorization, processing, and logic of the application. Spring Boot provides RESTful API endpoints for the processing of requests involving user management, customer data, projects, tasks, analytics, or event management. Spring Security along with JSON Web Tokens is used as part of the implementation for handling role-based security.

- Data layer Technology

The data layer utilizes MySQL as the relational database management system. The database schema is designed to provide structured relationships for business entities such as users, roles, clients, projects, and tasks. The database relationships provide integrity because of careful relationships between the foreign and primary keys.

V. IMPLEMENTATION DETAILS

The key area of implementation of CURIEM is integrating customer management, project management, and task management into one web-based application. The application is developed with full-stack technology for scalability and security reasons.

5.1 Frontend Implementation

The front end of the system has been developed with React, which facilitates creating a responsive UI. The application has been built with a component-based architecture to make it modular and reusable. Redux has been used as a state management tool in the application to manage sessions, role-based views, and common application data in a uniform way.

The user interface is made up of user dashboards for roles, forms for CRUD operations, and data display components like charts and tables. User role routing helps facilitate the ability for only authorized users to view functionalities linked to their roles.

5.2 Backend Implementation

It is Spring Boot-based in terms of backend technology. There is a full-featured framework in Spring Boot. There is business layer separation in controllers, services, and repositories. It is based on Spring Data JPA. Spring Data JPA simplifies data persistence and query management.

Security measures are implemented via the use of the Spring Security module with JWT authentication. The user credentials are encrypted, and token-based authentication provides safe access to the secured REST endpoints. The REST API provides functions for all major operations such as user management, customer record storage, project manipulation, task allocation, analytics, and event scheduling.

5.3 Database and Data Management

The system relies on a MySQL relational database. Database organization is accomplished through a relational database, with a carefully laid out schema that is capable of exhibiting relationships among primary entities like users, roles, and also other entities such as projects and tasks. Many-to-many and one-to-many relationships are facilitated through join tables.

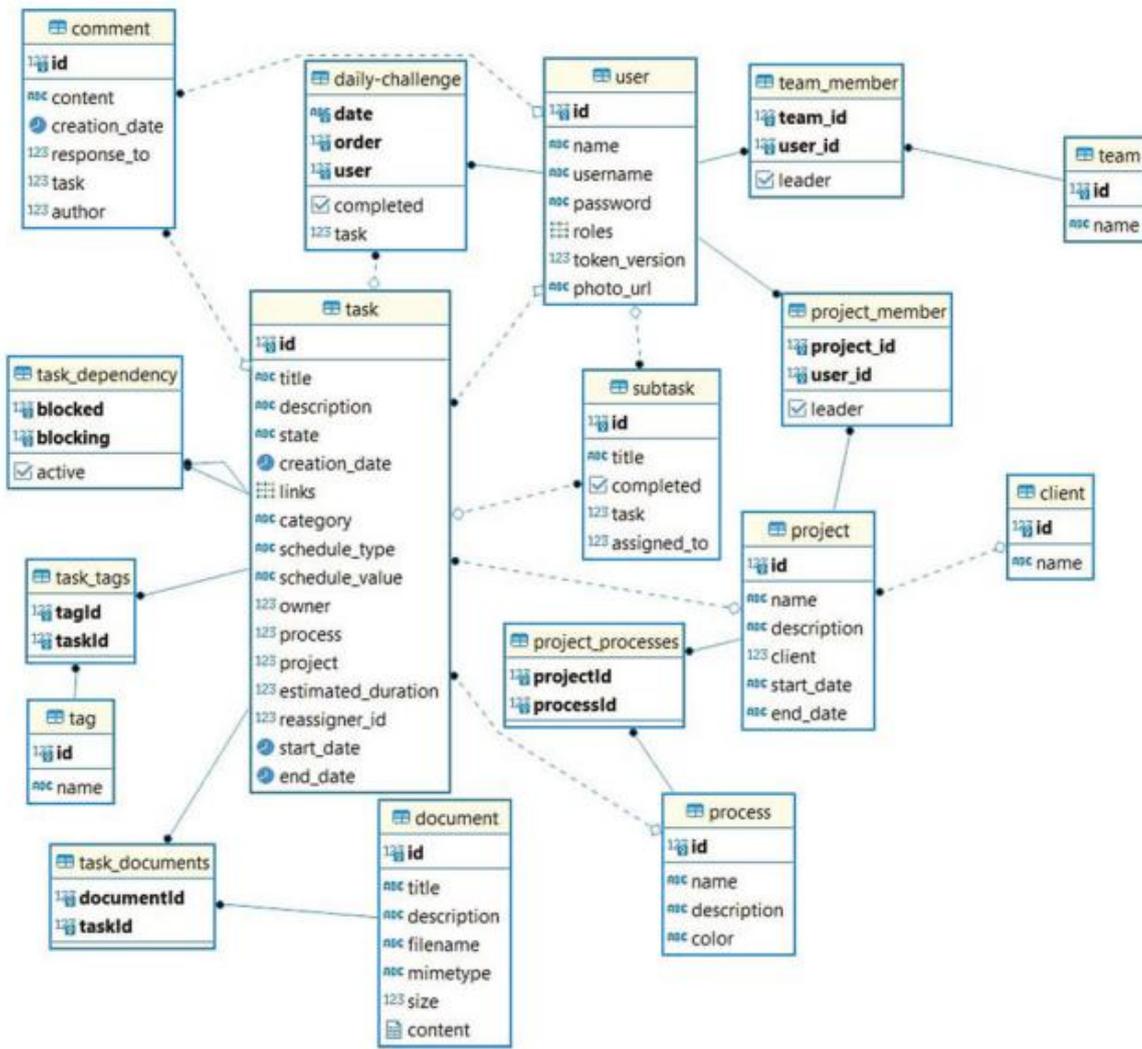


Fig. 12 Relational model of the task manager prototype

VI. RESULTS AND EVALUATION

The evaluation of the CURIEM system focuses on **functional correctness, system performance, and usability**. The purpose of this evaluation is to verify that the implemented features operate as intended and support the core objectives of the CRM application.

6.1 Functional Validation

All major modules of the system were tested across different user roles to ensure correct behavior. The system successfully supports user authentication, role-based access control, customer management, project creation, task assignment, and task status updates. Administrative users were able to manage users and roles, while managers and employees accessed only the functionalities permitted by their assigned roles, confirming proper enforcement of access control.

6.2 Performance and Reliability

Backend services implemented using Spring Boot demonstrated consistent response times for standard CRUD operations under normal usage conditions. REST API endpoints were tested using Postman to validate request handling, authentication flow, and error responses. Database operations maintained data consistency during concurrent access scenarios.

6.3 Usability Assessment

The React-based frontend provides a responsive and intuitive interface. Dashboard views present summarized project and task information, enabling users to monitor progress efficiently. The modular UI design simplifies navigation and reduces user effort in performing CRM-related tasks.

Overall, the evaluation results indicate that CURIEM is a reliable and functional CRM web application suitable for small and medium-scale organizations.

VII. CONCLUSION

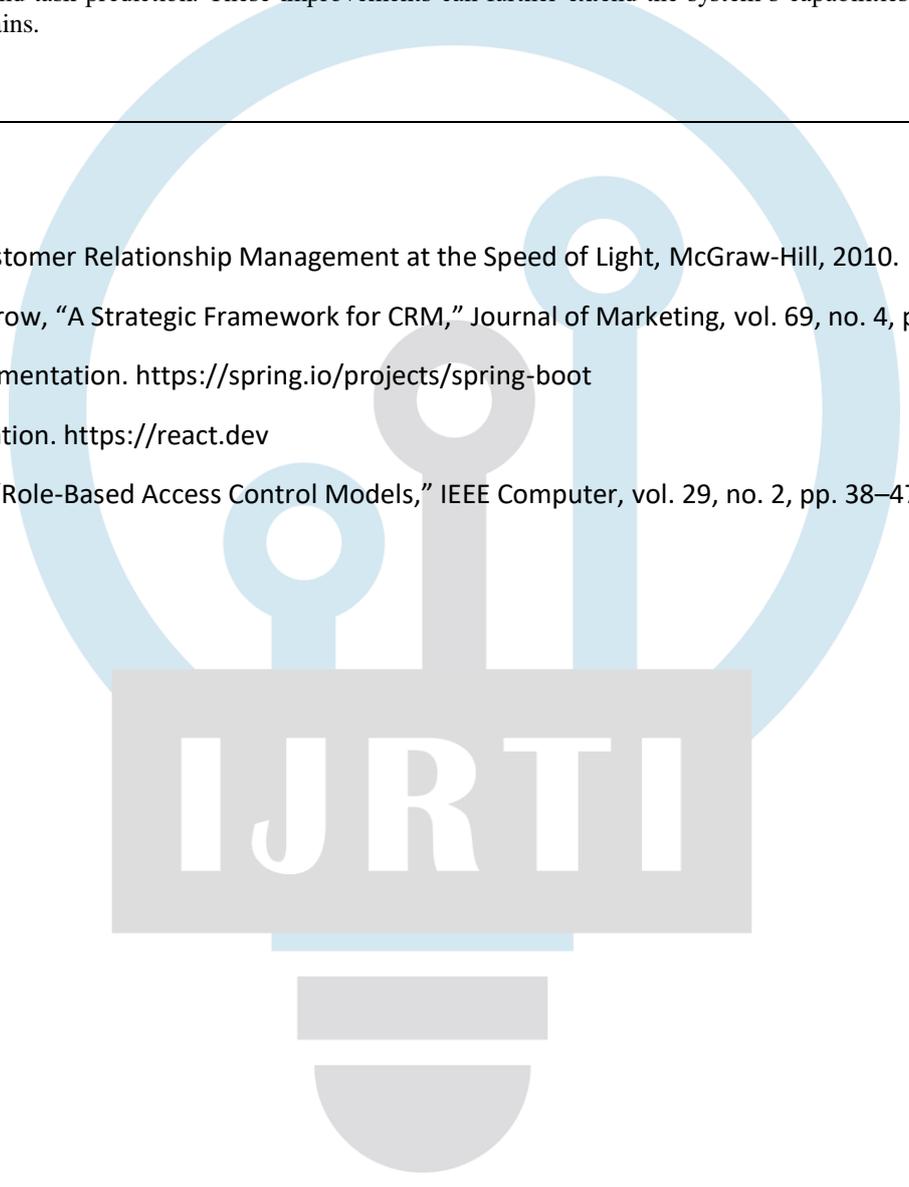
The presented paper discussed CURIEM, the web-based Customer Relationship Management system developed using Spring Boot and React. It integrates customer management with project and task tracking, role-based access control, and real-time dashboards within a single platform. CURIEM advances these features to improve workflow coordination, consistency in data, and operational visibility by addressing the limitations that exist within current CRM tools for small and medium-scale organizations.

The modular architecture and secure implementation will provide scalability, maintainability, and reliable system performance. Functional evaluation confirms that the system supports essential CRM operations effectively and meets the intended design objectives.

Future enhancements may include integration of advanced analytics, mobile application support, and machine learning techniques for customer insights and task prediction. These improvements can further extend the system's capabilities and applicability across different business domains.

REFERENCES:

- [1] P. Greenberg, *Customer Relationship Management at the Speed of Light*, McGraw-Hill, 2010.
- [2] A. Payne and P. Frow, "A Strategic Framework for CRM," *Journal of Marketing*, vol. 69, no. 4, pp. 167–176, 2005.
- [3] Spring Boot Documentation. <https://spring.io/projects/spring-boot>
- [4] React Documentation. <https://react.dev>
- [5] R. Sandhu et al., "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

A large, light blue watermark logo is centered on the page. It features a stylized lightbulb shape with a circular top and a rectangular base. Inside the circle, there are three vertical lines of varying heights, each ending in a small circle. The letters 'IJRTI' are printed in a bold, white, sans-serif font across the middle of the rectangular base of the lightbulb.

IJRTI