

Collaborative Task Management System

1. Introduction

- a. Project **Overview**: The Collaborative Task Management System is a distributed application designed to facilitate efficient task creation, assignment, and tracking among users. The system will be implemented using the Jolie programming language, leveraging its microservices architecture to ensure scalability and modularity.
- b. Objectives: The primary objectives of the project include:
 - i. Create a **user-friendly interface** for task management.
 - ii. Implement **microservices** for task creation, assignment, status tracking, and notifications.
 - iii. Enable **concurrent processing** to handle multiple users and tasks simultaneously.
 - iv. Ensure **secure communication and data storage**.

2. System Architecture

- a. **Microservices**: The system will consist of the following microservices:
 - i. **User Management Service**: Responsible for user registration, authentication, and profile management.
 - ii. **Task Service**: Manages the creation, assignment, and status tracking of tasks.
 - iii. **Notification Service**: Sends real-time notifications for task updates to users.
 - iv. **Concurrent Processing Service**: Handles concurrent user interactions and task updates.
- b. **Communication**: Communication between microservices will be implemented using Jolie's built-in communication primitives. The services will interact through well-defined interfaces, ensuring seamless integration.

3. Functional Requirements

- a. **User Registration and Authentication**:
 - i. Users can register with the system, providing necessary details.
 - ii. Secure user authentication mechanism.
 - iii. User profile management for updating personal information.
- b. **Task Management**
 - i. Users can create tasks, specifying details such as title, description, and due date.
 - ii. Tasks can be assigned to other registered users.
 - iii. Real-time status updates for tasks (e.g., in progress, completed).
 - iv. Users can view and filter tasks based on various criteria.
- c. **Notifications**
 - i. Real-time notifications for task updates.
 - ii. Notification preferences for users (e.g., email, in-app notifications).
- d. **Concurrent Processing**
 - i. Concurrent handling of user interactions and task updates.
 - ii. Efficient resource utilization to handle multiple users simultaneously.

4. Non-functional Requirements

a. **Security**

- i. Implement encryption for communication between microservices.
- ii. Secure storage of user data and task information.

b. **Scalability**

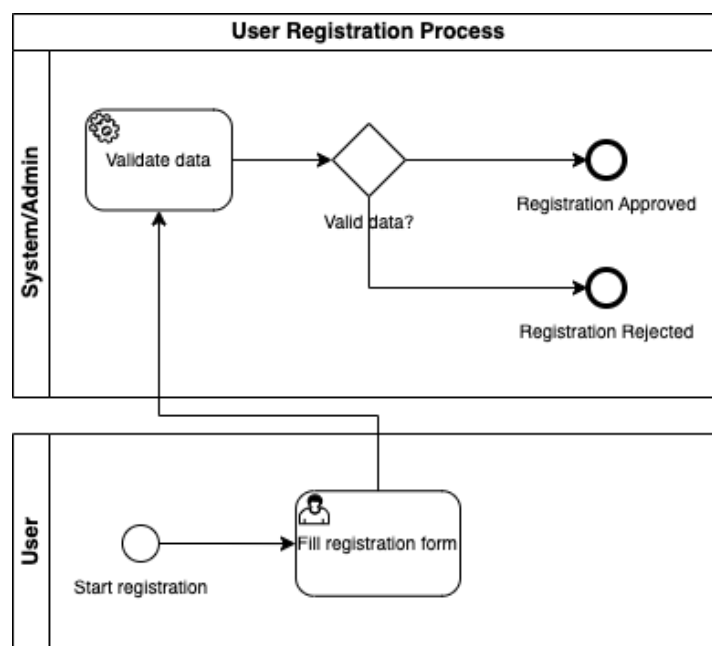
- i. Design microservices to scale horizontally to handle an increasing number of users and tasks.

c. **User Interface**

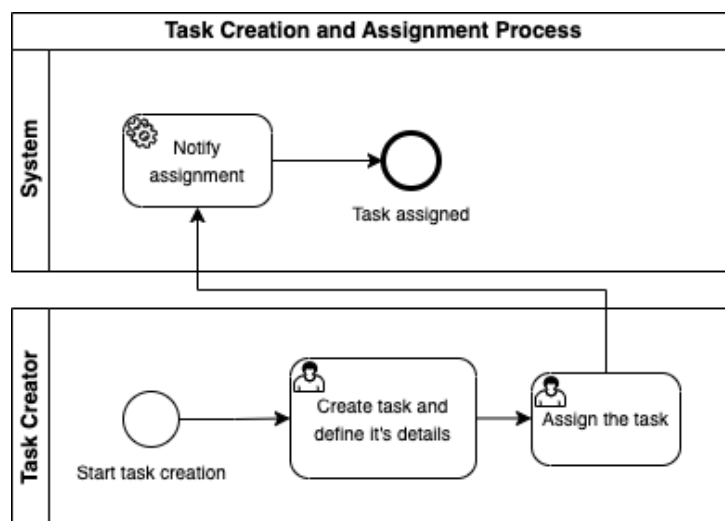
- i. Intuitive and responsive user interface for task management.
- ii. Compatibility with common web browsers.

5. BPMN diagrams of main processes

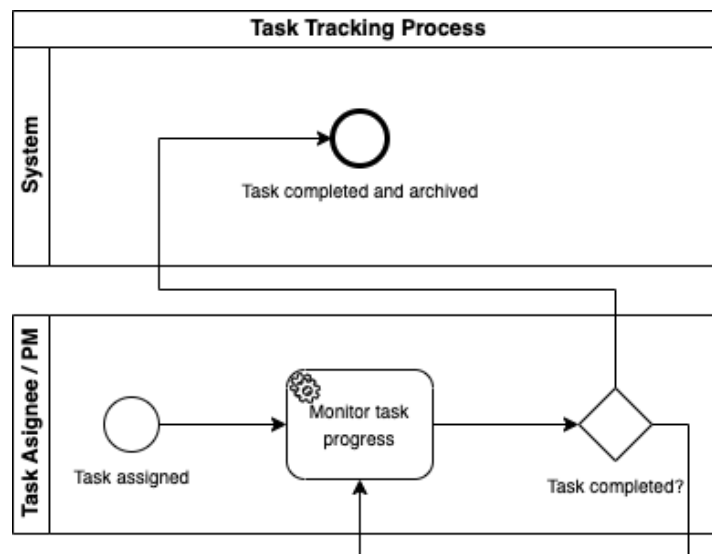
a. User Registration Process



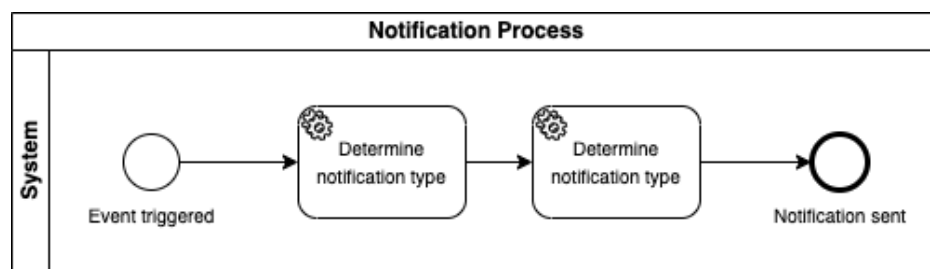
b. Task Creation and Assignment Process



c. Task Tracking Process



d. Notification Process



6. External Dependencies

- a. Utilize external libraries or frameworks for implementing specific features (e.g., user interface components).

7. Conclusion

- a. The Collaborative Task Management System aims to provide a robust and scalable solution for efficient task management in a collaborative environment. By leveraging the microservices architecture of Jolie, the system will be capable of handling concurrent user interactions while ensuring the security and integrity of user data.