OCTOBER 10, 2024

APPLIED RESEARCH ON HOW TO IMPLEMENT A SYSTEM TO OPTIMIZE EFFICIENCY AND SECURITY

RESEARCH PAPER

SANCHEZ, FRANCISCO; THOMAS, GILTON; PESTANA, CRISTIANO;
WERNECK ROALE, MIGUEL; LE, MINH
FONTYS UNIVERSITY OF APPLIED SCIENCE
Rachelsmolen 1, 5612MA, Eindhoven

TABLE OF CONTENTS

1. MAIN RESEARCH QUESTION

The project aims to develop a full-stack office reservation system that optimizes workspace usage and enhances flexible work arrangements by implementing core features such as room reservations, room management, work scheduling, team management, and administrative control. To determine the optimal way to integrate these functions into a cohesive system, the team has chosen the main research question that is guiding this project:

- **How to implement an office reservation system to optimize efficiency and security while supporting room reservations, work scheduling, team management, and administrative control?**

This question is essential because it will help the team determine the optimal way to integrate these functions into a cohesive and scalable system. By answering this question, the team aims to provide an effective solution that allows the organization to efficiently manage their workspaces and teams.

2. SUB-QUESTIONS

1. **How can encryption techniques be employed to protect sensitive information collected?**

- Spring Boot (Back-end): Implement encryption methods to secure sensitive data such as personal user information. Ensure data is encrypted both at rest (in the database) and in transit (using HTTPS).
- React (Front-end): Ensure secure transmission of data using HTTPS and handle encryption and decryption processes as needed for any sensitive user data.

2. **What methods are there for implementing a real-time notification system**?

- Spring Boot (Back-end): Use WebSockets or an event-driven architecture to send real-time notifications about changes, such as new reservations or cancellations.
- React (Front-end): Implement real-time notification features using components like Toast or modal popups to instantly notify users of booking updates or other important information.

3. **What measures can be implemented to ensure optimal performance when storing and retrieving data?**

- Spring Boot (Back-end): Use efficient data structures and optimize database queries to improve data retrieval performance.
- React (Frontend): Create forms where users can set their work schedule and sync them with the backend.

4. **How can continuous integration (CI) and continuous deployment (CD) practices be implemented to automate the testing, build, and deployment processes of the software solution?**

- Spring Boot (Back-end): Use GitLab CI to set up CI/CD pipelines that automate the testing, building, and deployment processes of the back end.

- React (Front-end): Automate the deployment of the front-end using CI/CD pipelines, including unit tests and integration tests to ensure reliability at each stage of development.

5. **How can room availability be maximized and booking conflicts avoided?**

- Spring Boot (Back-end): Implement scheduling algorithms and use database transactions to ensure that room booking conflicts are handled properly, preventing double bookings.
- React (Front-end): Create a real-time availability checker that dynamically updates to show available rooms and prevent users from selecting rooms already booked by others.

6. **How can the email system improve communication for employees with different work patterns?**

- Spring Boot (Back-end): Integrate email services like JavaMail or third-party services like SendGrid to automate email notifications and reminders for bookings.
- React (Front-end): Show in-app notifications and integrate with calendar APIs to remind users of bookings.

7. **How can single sign-on (SSO) be added to the system?**

- Spring Boot (Back-end): Implement SSO using OAuth 2.0 or OpenID Connect to allow users to securely authenticate on the system using a single login.
- React (Front-end): Integrate authentication libraries to handle SSO, allowing users to log in with their existing credentials seamlessly across platforms and services.

3. RESEARCH METHODOLOGY

3.1 Overview of the DOT Framework

The research focuses on developing a product rather than creating new scientific knowledge. To structure the research effectively, the team will use the Development Oriented Triangulation (DOT) framework. The DOT framework helps the team to organize the research activities and communicate the findings clearly.

The ICT project requires a combination of various ICT research methods. The team will use the toolkit available at ICT Research Methods to select appropriate methods for the study. The selected methods are categorized into four main areas: Library, Field, Lab, Showroom, and Workshop. Below is a detailed explanation of each category and the methods chosen.

3.2 Research methods

- **Library**
    - **Community Research:** Community research is using online community knowledge to find answers, check solutions, and share what you learn.
    - **Available Product Analysis:** Available product analysis checks if your project already exists, helping you decide whether to recreate, buy, or use existing solutions.
- **Field**

  - **Domain Modelling:** Domain modelling maps out key concepts and their relationships within a specific area to ensure clear understanding, often using techniques like UML class diagrams. It involves consulting stakeholders and developing a conceptual model to analyze the domain before progressing with a project.
  - **Explore User Requirements:** Exploring user requirements involves talking to users to understand their needs and how they will use the system. This process includes creating scenarios, use cases, and user stories to capture expected behavior and iteratively refining them with stakeholder feedback.
  - **Problem Analysis:** Problem analysis involves understanding a problem thoroughly before attempting to solve it. This includes asking questions like who, what, why, when, where, and how, involving domain experts, and ensuring clarity to avoid solving the wrong issue.
- **Lab**
  - **Component Test:** Component testing involves testing a subsystem or component in isolation to ensure it works correctly before integrating it with other parts. This includes checking the component's input and output against expected results using a model, simulation, and evaluation.
  - **System Test:** System testing checks the whole system to ensure it meets its requirements before going live. It involves creating a test plan with expected outcomes, running multiple test rounds, and comparing the results to find and fix bugs.
  - **Unit Test:** Unit testing involves checking individual parts of the code, like methods or functions, in isolation to find bugs early and ensure code stability after changes. This requires defining tests for each part, focusing on important cases, calculations, and exceptions, often using modern IDE tools.
  - **Usability Test:** Usability testing finds and fixes user issues before a system goes live. It uses prototypes or actual solutions to test interfaces, performance, and typical tasks, often with users thinking aloud and using automatic measurement tools.
- **Workshop**
  - **Brainstorm:** Brainstorming generates and develops new ideas by bringing people together to spark creativity. Use creativity techniques, accept all ideas without early criticism, and build upon each other's ideas.
  - **Code Review:** Code reviews help improve code quality by having peers identify bugs and suggest improvements. This can be done through team sessions, pair programming, or requiring reviews for each commit, ensuring adherence to coding standards and continuous feedback.
  - **IT Architecture Sketching:** IT architecture sketching helps define complex IT architecture by facilitating discussions among software designers and architects. Gather around a whiteboard, sketch high-level designs, and focus on important details only when necessary.

By employing these methods, the team aims to develop a comprehensive and effective software solution that meets the specific needs of an office reservation system.

4. RESEARCH METHODS AND CLARIFICATIONS

1. **How can encryption techniques be employed to protect sensitive information collected?**

- **Research methods:** Community Research, Domain Modelling, Component Test, Code Review.
    - **Community Research:** Investigate widely used encryption methods (e.g., AES, RSA) by engaging with developer communities and researching industry standards.
    - **Domain Modelling:** Map out which data elements are sensitive and need encryption and how the encryption/decryption process integrates with the system.
    - **Component Test:** Test the encryption and decryption process at the module level to ensure that each component functions correctly and securely.
    - **Code Review:** Ensure that encryption is implemented according to best practices, with a focus on security and performance.
- **Why:** Encryption protects sensitive data, such as passwords and personal information. Using these methods ensures that encryption is properly integrated and secure, with sufficient testing to prevent vulnerabilities.

2. **What methods are there for implementing a real-time notification system?**

- **Research methods:** Community Research, Code Review.
    - **Community Research:** Investigate common real-time technologies (e.g., WebSockets, server-sent events) used by other systems to provide real-time notifications.
    - **Code Review:** Ensure that the notification system code is clean, follows best practices, and is optimized for real-time performance.
- **Why:** Real-time notifications are essential for user engagement and responsiveness. These research methods help ensure that the system is designed for real-time efficiency, and user-friendly.

3. **What measures can be implemented to ensure optimal performance when storing and retrieving data?**

- **Research methods:** Community Research, Domain Modelling, Component Test, System Test, Unit Test, IT Architecture Prioritization.
    - **Community Research:** Explore how other systems achieve performance optimization (e.g., caching, indexing, query optimization).
    - **Domain Modelling:** Model the data flow and storage architecture to identify bottlenecks or inefficiencies in data handling.
    - **Component Test:** Test individual components (e.g., database queries, caching) for performance under various conditions.
    - **System Test**: Perform end-to-end tests to validate overall system performance under real-world conditions.

  - **Unit Test**: Ensure that individual data storage and retrieval functions operate efficiently.
  - **IT Architecture Prioritization**: Prioritize architecture decisions (e.g., database structure, partitioning) to improve performance in critical areas.
- **Why**: Optimizing data storage and retrieval ensures system responsiveness, especially under load. These methods help identify, test, and implement strategies that improve performance without compromising functionality.

4. **How can continuous integration (CI) and continuous deployment (CD) practices be implemented to automate the testing, build, and deployment processes of the software solution?**

- **Research methods:** Community Research, Brainstorm.
  - **Community Research**: Investigate tools and best practices for automating CI/CD (e.g., Jenkins, GitLab CI) used in similar projects.
  - **Brainstorm**: Collaborate with team members to design an effective CI/CD pipeline that integrates automated tests, builds, and deployments efficiently.
- **Why**: CI/CD practices reduce the risk of errors and accelerate the development process by automating repetitive tasks. Researching tools and brainstorming with the team ensures that the pipeline is tailored to the project's needs.

5. **How can room availability be maximized and booking conflicts avoided?**

- **Research methods:** Domain Modelling, Component Test, System Test, Code Review, IT Architecture Sketching.
  - **Domain Modelling**: Analyze the reservation and availability process to identify potential conflict points and optimize room usage.
  - **Component Test**: Test individual modules that handle room availability and conflict resolution.
  - **System Test**: Validate the system's ability to handle booking requests while preventing conflicts.
  - **Code Review**: Ensure that booking logic is robust and follows best practices.
  - **IT Architecture Sketching**: Design the system architecture to optimize room availability and efficiently handle reservations.
- **Why**: Avoiding conflicts and maximizing room availability is key to the system's usability. These research methods help thoroughly test the booking system, ensuring that it is efficient, reliable, and easy to use.

6. **How can the email system improve communication for employees with different work patterns?**

- **Research methods:** Community Research, Available Product Analysis, Explore User Requirements, Problem Analysis, Brainstorm.
  - **Community Research**: Investigate how other companies enhance email communication for flexible work schedules.
  - **Available Product Analysis**: Analyze existing email systems and features that support remote or hybrid work environments.
  - **Explore User Requirements**: Understand the specific communication needs of employees with different work patterns.

  - **Problem Analysis**: Identify common communication issues, such as email delays or information overload, and devise solutions.
  - **Brainstorm**: Collaborate on ideas for improving email communication, such as automated reminders or group messaging features.
- **Why**: Effective communication is crucial for remote or hybrid teams. These research methods help analyze the current system and identify improvements tailored to employee needs.

7. How can single sign-on (SSO) be added to the system?

- **Research methods:** Community Research, Available Product Analysis, Brainstorm.
  - **Community Research**: Investigate available SSO protocols (e.g., OAuth, SAML) and understand how other systems integrate SSO.
  - **Available Product Analysis**: Compare existing SSO providers and solutions to identify the most suitable one for the system.
  - **Brainstorm**: Collaborate on how to implement SSO in a way that balances user convenience and security.
- **Why**: SSO simplifies the login process for users while enhancing security. These methods help ensure a well-researched and efficient SSO integration into the system.