Department of Computer Science and Software Engineering (CSE)

Software Process - Soen 341

Project Report - Sprint 1

Team: Bala 2sm

12/02/2024

Table of Contents

1. Project Overview:

1.1 Project Objectives:

The primary objectives of the car rental website are:

- Facilitate a seamless and user-friendly platform for customers to rent vehicles for both short and long durations.
- Streamline the reservation process, making it efficient and accessible.
- Enhance customer experience through a comprehensive vehicle catalog and reservation system.
- Provide a centralized system for customer service representatives to efficiently manage check-in, check-out, and payment processes.
- Enable system administrators to perform essential CRUD operations on vehicles, user accounts, and reservations for effective management.

1.2 Scope:

The project scope includes the development of a web-based car rental application with the following key features:

- User Account Login: Each user, including customers, customer service representatives, and system administrators, can log in to their respective accounts to access personalized features and information.
- Vehicle Browsing: A catalog showcasing available rental vehicles suitable for both business and leisure travel.
- Reservation Initiation: Customers, including business travelers and tourists, can start a reservation by specifying location, pickup, and return dates, with options for vehicle type, category, and price range.

- Reservation Management: Customers can view, modify, and cancel reservations, accommodating the dynamic travel plans of business travelers and tourists.
- Additional Services: During reservation, customers can add extra services at an additional price, addressing the specific needs of business travelers and tourists.
- Branch Locator: Customers, particularly tourists, can find the nearest rental branch using postal codes or airports in unfamiliar locations.
- Rating and Review: Customers, including business travelers and tourists, can provide feedback and ratings for rented vehicles and the overall rental experience, assisting future travelers in making informed decisions.
- Check-in and Check-out Processes: Customer service representatives can efficiently manage reservations, providing tailored assistance for business travelers and tourists, ensuring a smooth rental experience.
- Reminder System: Implement a reminder system to notify users about upcoming reservations, important dates, and times, enhancing communication and ensuring users stay informed.

1.3 Target Audience:

The intended users of the car rental application include:

- Local customers: Individuals looking to rent vehicles for short or long periods, seeking a user-friendly and efficient reservation process.
- Business travelers: Professionals on business trips seeking efficient and reliable transportation solutions for short-term use.
- Tourists: Travelers exploring new destinations, looking for convenient and flexible rental options during their stay.

2. Project Approach:

2.1 Development Methodology:

Agile Methodology:

Justification:

The Agile development methodology is particularly advantageous for a small team working within a tight timeframe. In the context of this project, where completion is required within 10 weeks, Agile's ability to divide tasks into small, manageable increments becomes crucial. The iterative nature of Agile allows the team to continuously reassess priorities, ensuring that the most critical features are addressed first. Moreover, the regular feedback loops in Agile provide opportunities for quick adjustments, making it well-suited for time-sensitive projects with limited resources. The collaborative aspect of Agile also ensures effective communication within the small team, fostering a cohesive and efficient development process.

2.2 Project Timeline:

Milestones and Deadlines:

- Project Kickoff (Week 1): Initiate the project, define objectives, and establish team roles.
- Requirements Gathering (Weeks 2-3): Gather detailed requirements, including any additional functionalities or user roles. Those will be the results of Sprint 1 and should be handed in on February 12th.
- System Design and Architecture (Weeks 4-6): Develop the overall system architecture and design user interfaces based on gathered requirements. Those will be the results of Sprint 2 and should be handed in on March 11th.

- Development (Weeks 7-8): Begin iterative development cycles, focusing on core features such as user account management, vehicle catalog, reservation processes, and the reminder system. Those will be the results of Sprint 3 and should be handed in on March 25th.
- User Testing (Weeks 9): Conduct user testing to gather feedback and identify potential improvements.
- Deployment (Week 10): Deploy the application for initial use, ensuring stability and performance. Those will be the results of Sprint 4 and should be handed in on April 10th.

2.3 Collaboration and Communication:

Communication Channels:

Regular team meetings will be conducted weekly to discuss progress, address challenges, and plan upcoming tasks. In addition, more meetings might be conducted should the need arise. There will also be meetings when needed between people working on similar tasks (eg. Front-end, back-end, etc.)

A dedicated project communication private group on Moodle will be utilized for communications and also in-person meetings at the library.

Collaboration Tools:

Version Control: GitHub will be used for version control to track changes, and manage collaborative development.

Project Management: GitHub issues and tasks will help with tracking, sprint planning, and overall project management.

3. Technology Stack

3.1 Backend Framework

3.1.1 Framework A: SpringBoot

- Description:

  Spring Boot is an open-source Java-based framework, used to create production-grade applications with minimal configuration. Built on the Spring framework and provides a simple and fast way to run Spring applications.

- Rationale:
  - Justification for choosing Framework A.
    - Community support: it has a large community worldwide which contributes to its rich libraries and tools. Developers have access to plenty of resources, solutions, and best practices, which reduce the development and learning time.
    - Scalability: its modular architecture enables seamless scalability that will allow the website to adapt to the increasing requirements and grow in complexity between each sprint.
    - Ease of integration: with various other frameworks and third-party libraries, including OpenAPI and the Spring libraries.

- Qualitative Assessment:
    - Strengths:
        - Fast development: SpringBoot provides auto-configuration which allows developers to SpringBoot a project quickly and focus more on the logic.
        - Production-ready features: these are included in SpringBoot and they simplify managing and monitoring the website.
    - Weaknesses:

      due to the auto-configuration feature, SpringBoot might add some overhead to the application which might require extra optimization in performance-sensitive apps.
    - Use Cases:
        - Web applications.
        - API development.
        - Enterprise applications.

3.1.2 Framework B: NodeJS

- Description:

  NodeJS is an open-source, server-side JavaScript runtime environment, that allows developers to run JavaScript code outside of a web browser, which enables the building of high-performance web applications. It uses an event-driven, non-blocking I/O model which makes it lightweight and efficient.

- Rationale:
  - Justification for choosing Framework B.
    - Performance: the non-blocking I/O model allows NodeJS applications to handle constantly changing requests, and is suited for building real-time applications that require multiple connections. It is lightweight due to the optimized JavaScript engine.
    - Security: it provides security features such as a built-in mechanism for preventing vulnerabilities such as cross-site scripting and SQL injection. It has a vast ecosystem of security tools and libraries to help developers.
    - Maintenance: NodeJs is maintainable due to its simplicity, modular architecture, and npm packages, which provide access to thousands of reusable modules allowing code reusability, and test-driven development which contribute to easier maintenance.

- Qualitative Assessment:
  - Strengths:
    - Performance: known for its high performance and speed due to its non-blocking I/O model and event-driven architecture that contribute to fast response times.
    - Vaste resources: its popularity contributes to the availability of many resources and documentation, which allows easier development and integration.

- Weaknesses:
    - Unstable API: those are undergoing active development which can cause them to be unstable and might require manual updates.
    - Single-threaded nature.
- Use Cases:
    - Web applications.
    - Data streaming applications.
    - Single-page applications.
    - High-traffic websites.

3.1.3 Framework C: .NET

- Description:

  The .NET framework is a software development platform developed by Microsoft that allows the building of window-based applications. It includes a Common Language Runtime (CLR) which provides a managed execution environment for running .NET application.

- Rationale:
  - Justification for choosing Framework C.
    - Community support: it has a large active community of active developers that provide extensive documentation, forums, and third-party libraries, this ensures that new developers have access to the best resources and practices, which encourages better development.
    - Learning curve: .NET provides a manageable learning curve, especially for developers familiar with C# and Java. it provides similar syntax and comprehensive documentation and tool support in Visual Studio. In addition, Microsoft provides official resources and tutorials for different learning paths and skill levels.
    - Extensibility: it's highly extensible, it is customizable and its capabilities can easily be extended to meet the requirements, due to its class libraries, API handling, and NuGet packages.

- Qualitative Assessment:
  - Strengths:
    - Strong integration with Windows system, and other Microsoft technologies such as Azure cloud and SQL server which enhances working with the Microsoft System.
    - Security features: it has a built-in security mechanism to help build secure. applications, some of the features are, code access security, role-based security, and cryptography libraries. In addition, Microsoft constantly releases security updates to address vulnerabilities and threats.
  - Weaknesses:
    - Platform dependency: .NET is closely tied to the Windows platform. This may limit the deployment of applications especially when Windows isn't the primary operating system.
    - learning: it may be difficult to learn .NET without prior knowledge of OOP like C# and Java.
  - Use Cases:
    - Desktop applications.
    - Game development.
    - Mobile application.

3.2 Frontend Framework

3.2.1 Framework X: React

- Description:

  React is an open-source Javascript library for building user interfaces (UIs), where UIs are broken down into reusable and composable components, each with its logic. It was developed by Facebook. It allows the creation of interactive and dynamic web applications with component-based architecture.

- Rationale:
  - Justification for choosing Framework X.
    - User interface capabilities: React provides developers with powerful tools and libraries to create highly dynamic and interactive UIs.it allows for reusable UIs components due to its component-based architecture. All this allows developers to reuse and maintain the code and build large-scale applications with rich intuitive UIs.
    - Responsiveness: due to its virtual DOM, react enables excellent responsiveness and performance. Because it efficiently updates the components affected by changes in the application state, it has fast and seamless UIs updates which contributes to a smooth user experience and satisfaction.
    - Cross-browser compatibility: React is designed to be compatible with a wide range of modern web browsers, which ensures a consistent behavior and appearance across different platforms and devices. In addition to the React documentation that contains tools and libraries for handling browser compatibility questions.

- Qualitative Assessment:
  - Strengths:
    - Component-based architecture: it allows developers to create reusable code and modular UI components. This prompts code reusability and maintainability.
    - Virtual DOM: this minimizes the number of DOM manipulations resulting in a faster performance.
  - Weaknesses:
    - Tools complexity: React has a vast collection of tools and libraries, which can be overwhelming to new users and might require additional expertise to set up such as Babel, ESLint…
    - SEO challenges: React's single-page application can pose challenges on the search engine and will require additional considerations and optimizations in the server side.
  - Use Cases:
    - Single-page applications (SPA).
    - Progressive web applications.
    - E-commerce platforms.
    - Social media platforms

3.2.2 Framework Y: Angular

- <u>Description:</u>

  Angular is a popular open-source web application framework developed by Google. Designed for building dynamic and single-page web applications. It focuses on modularity, productivity, and testability. It follows a model-view-controller (MVC) architectural pattern.

- <u>Rationale:</u>
  - <u>Justification for choosing Framework Y</u>
    - Modularity: Angular's modularity allows developers to break down their applications into smaller, reusable components. With this self-contained UI elements with their logic, templates, and styles can be built in addition to facilitating the organization of application features and dependencies.
    - Performance: angular includes many performance optimizations, it provides techniques like ahead-of-time (AOT) compilation, which compiles templates into optimized JavaScript code, resulting in faster times. Its change detection minimizes unnecessary DOM use by only updating the necessary data which also optimizes performance.
    - Community support: Angular has a large community of active developers and contributors, this community provides extensive resources, tutorials, and documentation. This is how developers can find tutorials, and answers to their questions and collaborate with other developers to solve problems.

- Qualitative Assessment:
  - Strengths:
    - two-way data binding: it simplifies the synchronization of data between the model and the view, allowing the changes to the model to change the view automatically, this reduces boilerplate code.
    - Dependency injections: it provides comprehensive and manageable dependencies between different components and services, enabling easier unit testing.
  - Weaknesses:
    - Size: Angular’s bundle size is larger than most of the other frameworks which may impact load time and performance.
  - Use case:
    - Single-page applications.
    - Content management system.
    - Data visualization platforms.

3.2.3 Framework Z: Vue.js

- Description:

  Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. It is known for its simplicity, flexibility, and ease of integration, making it a popular choice for building modern web applications.

- Rationale:
  - Justification for choosing Framework Z.
    - Ease of integration: Vue is known for its ease of integration and adaptation into an existing codebase, allowing it to incrementally introduce Vue to an application without rewriting the whole codebase. It also allows easy integration with other libraries such as Vuex.
    - Component libraries: Vue provides rich documentation, pre-built UI frameworks, and component libraries. Such as Vuetify, Element UI, and BootstraVue, which have a lot of ready-to-use components like buttons, forms, styling components…
    - Developer experience: Vue prioritizes developer experience by providing a straightforward syntax, clear documentation, and a gentle learning curve. The syntax is inspired by HTML and JavaScript, which makes it easy to understand and convenient for beginner web developers.
- Qualitative Assessment:
  - Strengths:
    - Flexibility and adaptability: Vue is very flexible which allows it to be used in different project types, ranging from small beginner projects to large-scale applications, it can also be easily integrated into existing projects, or combined with other libraries providing the freedom to use any framework that suits the needs of the project.

- Weaknesses:
  - Smaller ecosystem compared to React and Angular, which are not as extensive and specialized.
  - It lacks the sponsor of a large corporation, while Angular is developed and maintained by Google and React by Facebook, Vue lacks this option which causes it to have a smaller community.
- Use case:
  - Single-page applications.
  - Social networking applications.
  - Data visualization dashboards.

4. Integration and Interoperability

4.1 Backend-Frontend Integration

To integrate the back end with the front end, we will create a REST API for the back end. We will then use Axios from the front end to make HTTP requests to our API.

4.2 Third-Party Services

- Google API: to allow the user to find the nearest branch through an interactive map.
- Maven: it's a tool used for dependency management. It automatically integrates all the required libraries in the project.
- Axios: is a tool used to make HTTP requests. We will use it to make GET and POST requests to our backend API.
- MongoDB: is used to store the databases.

5. Security Considerations

Measures we plan to take:

- We plan to use a built-in spring tool called spring security to make our authentication secure.
- Use the BCrypt protocol to encrypt usernames and passwords before storing them.

Some security considerations we would need to address in future sprints:

- Make sure that users can access only their reservations.
- Make sure that only the admin can do CRUD operations on users, cars, and reservations.

6. Conclusion

For the car rental website, the Agile approach was chosen to ensure flexibility and adaptability within the 10-week timeframe provided, it will allow iterative development cycles, while fostering effective collaboration and communication among team members. The technology stack chosen for the project includes Spring Boot, Node.js, and .NET for the backend. Meanwhile, for the front end, React, Angular, and Vue.js were chosen. Backend-frontend integration will be achieved through REST API for the communication, with third-party services like Google Maps API and tools like Maven and Axios, and finally MongoDB for the databases. Security measures taken include Spring Security for authentication and BCrypt for encryption, with consideration to adding access control and authorization. Finally, the frameworks in addition to the third-party services and the security measures were all chosen to prioritize performance, and developer experience ensuring a successful and smooth development environment and promoting a seamless learning curve, to build a successful car rental website.