

SOEN 341 Software Process (Hexad)

Team Project, Winter 2024

Project Title: A Car-rental Web Application

Appendix A. Project Approach and Technology Stack Selection Template

Car Rental Web Application

1. Project Overview

1.1 Project Objectives

The 10-week car rental web application project follows Agile Scrum methodology to create a user-friendly platform that facilitates people renting vehicles for short periods of time. The initial two weeks involve training on Agile Scrum and setting up GitHub for version control and bug tracking. The following sprints (3 to 4 weeks each) focus on developing core features, such as initiating reservations, browsing available vehicles, and implementing CRUD operations for users, vehicles, and reservations. Additional functionalities include a branch locator and a rating system to enrich the user experience.

1.2 Scope

The following sprints (3 to 4 weeks each) focus on developing core features, such as initiating reservations, browsing available vehicles, and implementing CRUD operations for users, vehicles, and reservations. Additional functionalities include a branch locator and a rating system to enrich the user experience.

1.3 Target Audience

The target audience includes a diverse range of users looking for a convenient and an efficient solution for their temporary transportation needs. On top of clientele users, there are client-representatives and system administrators. The user clients can start a reservation, browse vehicles available for rent, manage reservations and can provide feedback. The client-representative takes care of the check-in and check-out process for the clients, creating new reservations in the

system, confirming reservations, etc. Finally, system administrators do all the CRUD operations on vehicles, user accounts, and on reservations.

2. Project Approach

2.1 Development Methodology

The agile methodology was chosen for this project for various reasons. Firstly, agile allows us to divide the project into smaller tasks for each sprint. By doing so, we can focus more on each feature of our car rental website and ensure that everything is properly implemented. We can also receive frequent feedback from our TA and adapt accordingly. Additionally, this methodology emphasizes collaboration amongst team members. Indeed, daily scrum meetings ensure that everyone is progressing and it keeps us on the same page while weekly meetings are used to discuss and find solutions for various issues. Having these dedicated meeting times helps us keep up as everyone has different schedules. Also, like mentioned previously, frequent deliveries allow us to gather feedback from stakeholders, which in this case is our TA. This ensures that the final product will meet the customers needs and expectations. Lastly, agile encourages the assessment and management of risks. This allows us to address potential issues quickly and it reduces project delays or failures which is even more relevant to us as we have about 3 months to complete this project. To sum it up, iterative development, collaboration, customer satisfaction and risk management make agile the best development methodology for this project.

2.2 Project Timeline

Sprint 1: Feb 12th - Set up the GitHub for the project and documentation

Sprint 2: Mar 11th - Implement core features

- Start a reservation
- View/Modify/Cancel a reservation
- Browse vehicles for rent
- CRUD operations on users
- CRUD operations on vehicles
- CRUD operations on reservations

Sprint 3: Mar 25th - Improve core features by adding new options and complete any core feature that wasn't completely done or properly working

Sprint 4: Apr 10th - Add a special feature and improve front end visuals

2.3 Collaboration and Communication

Our main communication tool is discord where we set up a server for the project. We have a channel dedicated for our daily scrum and our weekly meeting. Then we have dedicated channels for specific topics such as front-end, back-end, testing, and pr-reviews. Lastly we also have a general channel for miscellaneous subjects. On the other hand, our main collaboration tool for task division and code is GitHub where our project is set up. We have our various tasks (6 user stories) and their detailed description available for everyone to see. We also have our logs where we detail the specific work we accomplished. Additionally, for each task, we have a dedicated branch that we merge into the main once it has been peer reviewed by at least one other teammate.

3. Technology Stack

3.1 Backend Frameworks

3.1.1 Express.js

- Description:

Express.js is a minimalistic and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is designed to create RESTful APIs with ease.

- Rationale:

Express.js is selected for its simplicity and scalability. it is particularly well-suited for the project due to its lightweight nature, making it easy to learn and quick to set up. Its middleware architecture allows for the seamless integration of additional functionalities, making it a suitable choice for handling diverse backend requirements such as the MSSQL database.

The framework has a large and active community, ensuring ongoing development, frequent updates and a large amount of resources for problem-solving and enhancements. While Express.js is known for its simplicity, it remains scalable, enabling the development of robust APIs that can handle potential future growth in both users and features. Finally, Express.js integrates seamlessly with various libraries and databases providing flexibility in choosing components that best suit the project needs.

- Qualitative Assessment:

Express.js possesses several strengths that make it a preferred choice for many web developers. One of its key advantages lies in its simplicity. The minimalist and easy-to-use framework reduces the learning curve for developers. Its straightforward design allows for rapid development and iteration, making it an excellent choice for projects where speed is crucial. Additionally, Express.js offers a robust middleware ecosystem, enabling developers to seamlessly integrate additional features without complicating the core codebase. This versatility and simplicity contribute to the framework's strength in delivering and maintaining the website.

While the framework has numerous strengths, it also comes with certain weaknesses. One notable aspect is its structure which promotes simplicity but limits architectural choices. Express.js tends to enforce a particular project structure, potentially causing challenges. Additionally, the more advanced concepts might be harder for beginners to learn. Despite these weaknesses, Express.js remains a powerful and widely used framework that can be effectively managed with a solid understanding of its principles.

Express.js finds extensive application in various scenarios due to its versatility. One notable use case is in API development. Its lightweight design and robust routing capabilities make it an ideal choice for creating RESTful APIs, allowing developers to handle HTTP requests efficiently. Another use case is in microservices architecture, with its ability to support modular and scalable application design. Lastly, Express.js is well-suited for prototyping, offering a quick and effective way to build and test web applications during the early stages of development.

3.1.2 Django

- Description:

Django is a high-level, open-source web framework for Python that encourages rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern emphasizing reusability and less code. Django includes an Object-Relational Mapping (ORM) system for database interaction, a templating engine, and a robust set of built-in features.

- Rationale:

Django is chosen as a backend framework for several compelling reasons. First and foremost, its vibrant and extensive community support is a key factor. The large and active Django community contributes to the framework's reliability, security, and ongoing enhancement. This ensures that developers have access to a wealth of resources, including documentation, forums, and third-party packages, facilitating smoother development workflows. Additionally, Django's scalability is noteworthy, making it a versatile choice for projects of varying sizes. Whether building a small

application or a large-scale system, Django provides the necessary tools for efficient development and scaling. Another crucial aspect is the framework's ease of integration. Its set of built-in features, such as the admin interface, authentication system, and ORM, simplifies the integration process.

- Qualitative Assessment:

Django has several strengths that make it a powerful choice for web development. The "batteries-included" philosophy equips developers with a comprehensive toolkit, minimizing the need for external dependencies. The built-in Object-Relational Mapping (ORM) system streamlines database interactions, simplifying data modelling. Additionally, Django's emphasis on security, evidenced protection against common vulnerabilities, ensures robust protection for applications.

While Django offers a plethora of benefits, it does have some limitations. The learning curve can be steep, particularly for beginners, due to the framework's richness and extensive feature set. Moreover, the "monolithic" nature of Django, characterized by its inclusiveness, may lead to larger codebases in certain scenarios, potentially impacting maintainability. Despite these challenges, the trade-off is often justified by the framework's productivity gains.

Django finds its forte in various application domains. For Content Management Systems (CMS), Django's admin interface facilitates content handling and customization. E-commerce platforms benefit from Django's scalability and built-in features, providing a solid foundation for transactional applications. In data-driven applications, Django's ORM and data modelling capabilities prove invaluable, ensuring seamless database interactions. These diverse use cases highlight Django's adaptability across different project types and sizes.

3.1.3 Feather.js

- Description:

Feather.js is a lightweight and highly modular web framework for building real-time applications. Designed with simplicity and flexibility in mind, Feather.js allows developers to create scalable and maintainable applications with built-in support for real-time functionality. It follows a microservices architecture and seamlessly integrates with various databases such as MSSQL, making it a versatile choice.

- Rationale:

Feather is a compelling choice due to its lightweight yet powerful nature. The framework's emphasis on simplicity ensures a streamlined development process, reducing the learning curve. Feather.js

excels in providing built-in support for real-time functionality, a crucial feature for our dynamic and interactive website. The microservices architecture facilitates modularity, enhancing scalability and maintainability as the project evolves. Additionally, the active and supportive Feather.js community, coupled with the framework's high extensibility, ensures that there is access to a growing ecosystem of plugins and the flexibility to tailor the framework. Overall, Feather.js offers an optimal balance of simplicity, real-time capabilities, and extensibility, making it a well-suited and efficient choice.

- Qualitative Assessment:

Feather.js brings several strengths to the website development, making it a good choice for certain scenarios. One of its primary strengths lies in its built-in support for real-time capabilities, making it well-suited for websites that require instant updates and collaborative features. Feather.js offers a streamlined development experience and makes it simple to learn.

While Feather.js provides a lightweight and modular approach to web development, it may have some limitations. Notably, its focus on simplicity may result in a lack of certain advanced built-in features compared to more comprehensive frameworks. Additionally, the community size, while active, might be smaller than that of larger frameworks, potentially impacting the availability of third-party plugins for specialized functionalities.

Finally, Feather.js is particularly well-suited for websites with specific needs. Its real-time capabilities make it an excellent choice for websites requiring features like live updates, chat functionalities of collaborative tools. the microservices architecture makes it adaptable for projects with scalability and maintainability requirements, providing a modular structure that aligns with the evolving needs of a growing website. Feather.js shines in scenarios where a lightweight and flexible framework is preferred, making it ideal for small projects where simplicity, real-time features, and modularity are key considerations.

3.2 Frontend Frameworks

3.2.1 React.js

- Description:

React is a declarative open-source Javascript library used for creating user interfaces. This software, created by Facebook, simplifies the process of creating UI components. It allows developers to create reusable UI components and efficiently manage them, which makes it one of the most popular libraries for web development and makes it easier to create interactive and dynamic web applications.

- Rationale:

React was chosen as a framework due to its strong emphasis on creating modular and reusable UI components. This requires less coding and overall complexity which allows for an easily maintainable and scalable codebase. When it comes to responsiveness, React's virtual DOM allows for efficient updates by calculating and applying only the necessary and optimized changes to the actual DOM. This increases performance and responsiveness, especially on applications with frequent updates. React was also designed to work seamlessly across all modern browsers such as Edge, Firefox, Chrome, Safari, etc. Additionally, this software also has a small learning curve and is easy to use compared to other comparable frameworks such as Angular.

- Qualitative Assessment:

One of the major strengths of React is that it is component-based and allows for more reusable, predictable and easier-to-debug code. Additionally, its virtual DOM is a valuable feature that enhances performance and does the optimization of the web application in the background. As mentioned previously, React is a very commonly used library meaning that there are plenty of resources and third-party tools that are widely accessible.

When it comes to weakness, React is a library that is constantly changing and being updated making it less stable than some other frameworks. This makes it harder for developers to keep up with it as they are forced to constantly learn new ways of writing their code. The constant updating also makes it hard for documentation to be up to date which results in a lot of poor documentation.

- Use Cases

React is often used when building single-page applications with dynamic and interactive UI or large-scale applications because of its modular structure and component reusability. Additionally, its efficient updating mechanism makes it great for the creation of real-time applications where responsiveness is crucial.

3.2.2 Vue.js

- Description:

Vue.js is an open-source progressive JavaScript framework for building user interfaces. Often praised for its simplicity, it allows developers to easily create and integrate interactive and reactive web applications. Vue.js is suitable for all kinds of projects no matter the scale or complexity as it was designed to be incrementally adoptable.

- Rationale:

Vue.js is a great framework to use because of its approachable learning curve and its flexibility. It excels in scenarios where rapid development and ease of integration are a priority such as during CI/CD (continuous integration/continuous deployment) practices. It is a lightweight framework that does not compromise on capabilities. Since Vue.js can be incrementally adopted, it can be used side by side with other frameworks allowing for more flexibility. Additionally, Vue.js simplifies the creation of dynamic interfaces through its reactive data binding and component-based architecture.

- Qualitative Assessment:

Vue.js' main strength is its ease of integration. It can be adopted into existing projects or side by side with another framework. This allows developers to pick and choose specific parts for their applications without needing to change their entire project. Vue.js also has an easy learning curve making it a very accessible framework for any kind of programmer. Finally, its reactive data binding makes creating reactive interfaces a lot more straightforward as the process of managing and updating UI based on data changes is very intuitive.

Vue.js is not as big of a framework as React or Angular, meaning that it has a smaller ecosystem. It is also not as popular as React which could mean that there is less community support. Finally, although the learning curve for the basic use of Vue.js is very approachable, it becomes steeper when trying to learn some of its more advanced features.

- Use Cases:

Vue.js is often used for prototyping or building smaller applications quickly because of its simplicity and ease of integration. Additionally, its reactive data binding and component-based structure make it suitable for the creation of single-page applications.

3.2.2 Next.js**- Description:**

Next.js is a React-based javascript framework, designed to build modern web applications. Although it can be used as a front-end and back-end framework we'll be focusing on its front-end features. Next.js simplifies the development of statically generated web applications. It provides a flexible environment for creating dynamic websites with React as well as built-in solutions for routing, server-side rendering, and code splitting.

- Rationale:

Next.js is great at creating dynamic and interactive user interfaces with React components. Its static site generation contributes to improved performance, SEO, and user experience. Next.js leverages the React library for building UI components, providing a component-based architecture that promotes a declarative and efficient approach to building user interfaces. Next.js also facilitates cross-browser compatibility by generating static HTML for pages, ensuring consistent rendering across different browsers.

- Qualitative Assessment:

One of Next.js' strengths is that it simplifies the setup and configuration process which allows developers to focus on building the features rather than managing complex configurations. The framework's built-in support for code splitting, a technique that loads only necessary JavaScript files for the current page, contributes to improved load time and overall performance. Next.js also provides efficient rendering options including server-side rendering and static site generation.

When it comes to its weaknesses, Next.js can have a bit of a sleeper learning curve for less experienced developers, especially for those unfamiliar with React. Additionally, Next.js follows certain conventions which may limit the customization possibilities for developers that like to have more control over configurations.

- Use Cases:

Next.js is a great framework for building websites that are content-driven and require efficient rendering. It is also well suited for e-commerce websites that require dynamic and interactive user interfaces and can leverage Next.js to provide a seamless and performant shopping experience. Finally, Next.js can also be used to create static websites due to its capability to pre-render pages at build time for optimal performance.

4. Integration and Interoperability

4.1 Backend-Frontend Integration

Backend

The chosen backend is JavaScript, MySQL, Node.js, Express.js, Jest and webpack. The strategy for integrating JavaScript is to use it as the programming language on which the functionality of the website relies. It will be handling core functionality, such as confirming a reservation.

MySQL will be integrated in every back end code related to CRUD operations. MS SQL will be integrated in every back end code related to CRUD operations and it was chosen over MYSQL due to its easier implementation.

Then to run JavaScript on the server side or on the chosen database, Node.js will be used (Ex: setting up a connection). The reason for choosing Node.js is because it can handle multiple requests concurrently and it's also fast to execute. Express.js will be implemented as well alongside Node.js to facilitate the integration of APIs and requests such as logging. Jest will be used for testing our unit tests and webpack for our client integration, as webpack efficiently manages multiple JS files by grouping them into one enhanced package.

Frontend

We decided on first working on an initial design of the website on Figma which translates the visual work into html and css code. Once the design is approved by the team, all the html and css code will be added to the react index file which will contain the main components of the html and a imported into the index react file a css file that contains the actual code of design (i.e. background color, font, font-color, font size, etc). This process was decided for the main page, other features' UI will be developed through figma after the feature's code is working correctly with the tests and is fulfilling all the requirements for the features. The reason behind this logic is to prioritize the working features instead of the design of a not fully developed feature.

4.2 Third-Party Services

- Google's Nearby Search API

To find the nearest branch from a postal code or an airport, the team is thinking of using Google's API "Nearby Search". Nearby Search . This free API appears to be accessible because the results of the API are either in XML or in JSON which can be easily worked with through Javascript. The API, Nearby Search, is called with a HTTP URL that looks like the following format : <https://maps.googleapis.com/maps/api/place/nearbysearch/output?parameters>. The required parameters for the API are location and radius. Which is exactly what the team needs for the project's feature of finding the nearest branch. More parameters can be added to the search such as but those are optional. The only con to this API is that it shows results of real location which would mean that the team would have to find a way to create a fake real location for the branches of the company. More documentation for the API can be found on this website below : <https://developers.google.com/maps/documentation/places/web-service/search-nearby>.

- Reviews.io API

Reviews.io is an API that makes reviewing very accessible for developers. This API can not only store reviews for either or both the product and the company, it also stores them with tags allowing easier search through the reviews. With this API, the company is able to invite through email or SMS customers to rate and review the company/product they ordered aka the car they rented. It is also possible to unsubscribe the customer from both invitations. Reviews.io's storage of the ratings and reviews can be used to retrieve review product ratings for features such as a review listing for products being sold or rented in the context of this project. The API can also retrieve basic stats like total reviews or overall score. It is also important to note that the review display can be fully customized through calls that are explained more in detail on the API's website : <https://developer.reviews.io> . Using this API for the reviewing and rating of the customer's experience would be a good tool to use to lessen the complexity of the task.

- **Figma**

This application allows teamwork on the design of a project to make sure the design is made in a commonly agreed on fashion. Not only does it look like a photoshop where multiple people can modify the project being worked on but it also translates the visual work done on the design of the web application into html and css code. This helps web developers when creating a website because it saves the process of trial and error on css to get the desired look on just a certain feature. Therefore, to save us time working on the frontend and to make sure the design is agreed on by the whole before going any further and meeting future issues of not having the same font or having a website that isn't the same through the pages, the team will be using this third party service application to create the UI of the website.

5. Security Considerations-Provide an overview of security measures and considerations for

both the backend and frontend:

In terms of the frontend, we can use HTTPs to ensure that the data transmitted between the users and the server is not intercepted by data thefts. Another point to consider for security would be to ensure that the browser does not send the user's information to another website by implementing security headers. As for the backend: enabling strong passwords for users on our database to protect their data. implement user input validation for instance validating email addresses using regular expressions to prevent phishing emails from entering the system.

6. Conclusion

All in all, our team's approach to this chosen project is to focus on delivering high quality functionalities in increments. We're also prioritizing core features such as setting up CRUD

Operations of the website first before dwelling into the trivial ones like deleting a user's account. As well, we are implementing the Agile methodology to manage the progress and the execution of our project. In order to provide high deliverables, we are using high end technologies for backend and frontend such as React, Node.js and Express.js. As for security concerns, we are aiming to ensure protection by means of the front and backend.