Technologies and Tools Used

Programming Languages

Java (version 17)

Java is the main programming language for the backend of the application. Version 17, which has long-term support (LTS), was chosen for its reliability, performance improvements, and support for new language features. Java is well-suited for building robust enterprise applications, and its use here ensures scalability, maintainability, and a large pool of available libraries and frameworks.

TypeScript (version 4.5)

TypeScript was used for the frontend development of the application. This language was chosen over JavaScript because it introduces static typing, which reduces runtime errors and helps catch bugs early during development. TypeScript also improves code clarity and refactoring, making it easier to maintain the codebase, especially in large projects.

Frameworks and Libraries

Spring Boot (version 2.5)

Spring Boot is a widely-used Java-based framework for building backend applications. It simplifies the configuration of complex enterprise applications, allowing developers to focus on writing business logic rather than boilerplate code. Spring Boot was chosen for its rapid development capabilities, built-in dependency management, and integration with modern databases and web services. Its modular structure also supports microservices architecture, making it easier to scale the backend system.

React (version 18)

React is a popular JavaScript library used for building user interfaces. It was selected for the frontend due to its component-based architecture, which allows for reusable and modular UI elements. React provides a virtual DOM, improving performance by minimizing updates to the actual DOM. This makes it well-suited for building dynamic, interactive web applications like the "Business Trip" system. Its widespread community support and extensive ecosystem of tools and libraries make it a solid choice for frontend development.

Tailwind CSS (version 2.2)

Tailwind CSS is a utility-first CSS framework used to style the frontend. It was chosen for its flexibility and the ability to create custom designs quickly without writing extensive custom CSS. Tailwind encourages the use of predefined classes that can be applied directly to HTML elements, enabling developers to create responsive and consistent user interfaces with minimal effort. Its lightweight nature also helps in maintaining a clean and organized codebase.

Vite

Vite is a modern build tool that was chosen to optimize the development and build process for the frontend application. Vite provides fast hot module replacement (HMR), which allows for instant updates during

development, enhancing the developer experience. Vite's optimized build system also improves performance, making it well-suited for large React applications.

Database

PostgreSQL (version 13)

PostgreSQL is an open-source relational database system that was selected for its reliability, scalability, and advanced features. It provides strong support for ACID (Atomicity, Consistency, Isolation, Durability) transactions, which are crucial for handling business-critical data like travel expenses and trip approvals. PostgreSQL also offers powerful querying capabilities and is widely used in enterprise environments due to its performance and ability to handle complex data relationships.

Development Tools

Visual Studio Code (version 1.62)

Visual Studio Code (VS Code) is a lightweight and powerful integrated development environment (IDE) that was used primarily for frontend development. VS Code was chosen for its speed, ease of use, and extensive marketplace of extensions, which support TypeScript and React development. It offers excellent debugging tools, version control integration, and code navigation, making it ideal for managing complex frontend projects.

IntelliJ IDEA (version 2021.2)

IntelliJ IDEA is a leading IDE for Java development and was used for backend development. IntelliJ provides advanced code refactoring, smart code completion, and powerful debugging tools. It integrates seamlessly with Java-based projects, making it a preferred choice for developers working on enterprise applications. Its built-in support for Spring Boot enhances productivity by automating many common tasks like dependency injection and service configuration.

Git (version 2.34)

Git is a distributed version control system that was used to manage the codebase. Git was selected for its ability to track changes, support branching and merging, and enable collaboration among multiple developers. GitHub, a cloud-based Git repository, was used for hosting the project, facilitating easy code review, pull requests, and issue tracking.

Testing Tools

JUnit (version 5)

JUnit is a widely used testing framework for Java, and it was utilized for backend testing. JUnit 5 was chosen for its powerful and flexible features, which support writing unit and integration tests. This ensures that backend components like controllers, services, and repositories function correctly. Testing with JUnit helps catch bugs early, improve code quality, and ensure that the backend logic adheres to expected behaviors.

Selenium (version 4.0.-beta.14)

Selenium is a powerful tool used for automating web browsers. It is widely used for functional testing of web applications to ensure they behave as expected across different environments. Selenium was integrated into the testing process for automating end-to-end (E2E) tests, verifying that user interactions and workflows in the application work seamlessly. It supports multiple browsers and allows for writing tests in various programming languages, making it an essential tool for ensuring high-quality, reliable web applications.

Deployment Tools

Docker (version 20.10)

Docker is a platform for containerizing applications, and it was used to package the project into containers for consistent deployment across environments. Docker was chosen because it allows developers to create isolated environments where the application, along with its dependencies, can run identically in development, testing, and production. This approach reduces the “works on my machine” problem and streamlines the deployment process by ensuring all environments are identical.

Cloud Platform

Render

Render is the cloud platform used to host the application. Render provides scalable computing power, storage, and networking services. It was selected for its ability to handle dynamic scaling, ensuring the application can accommodate an increasing number of users and requests without performance degradation. Render also provides high availability and redundancy, which is critical for an application that handles sensitive business data.

Explanation of Technology Choices

- **Java and Spring Boot** were selected for their maturity, security, and ability to handle complex business logic in a performant manner.
- **React and TypeScript** were chosen for the frontend because they offer a combination of dynamic UI building and strong typing, reducing the likelihood of bugs and improving maintainability.
- **PostgreSQL** was preferred for its ability to efficiently handle relational data and support complex queries.
- **Vite** was chosen for its speed and optimization of frontend development, especially with React applications, improving the build and development experience.
- **Selenium** is used to automate the browser and perform end-to-end tests, ensuring the application behaves correctly and providing automated regression testing for ongoing maintenance and new features.
- **JUnit** is a key testing framework for Java that ensures reliable and maintainable backend logic through unit and integration testing, helping developers verify that the backend system functions as expected under various conditions.

- **Docker** ensures that the development and production environments are consistent, reducing deployment issues.
- **Render** provides the scalability, availability, and reliability needed for hosting an application that manages business operations like travel and expense tracking.