**Documentation TaskTide**

- Turcut Stefan-Adrian –

**BackEnd:**

* *Spring Boot:*

A sophisticated Java framework tailored for building resilient and scalable server-side applications.

Leverages Java's capabilities for enterprise solutions, providing extensive libraries and tools.

Promotes convention-over-configuration, reducing boilerplate code and enhancing developer productivity.

* *Docker:*

Containerization technology enabling the packaging of applications and dependencies into portable containers.

Ensures consistent deployment across diverse environments, facilitating scalability and reproducibility.

* *MySQL:*

A reliable relational database management system known for its performance and stability.

Stores data in structured tables and supports SQL for querying and manipulation.

Widely embraced for its ACID compliance and seamless integration with Java applications.

* *Kafka:*

A distributed event streaming platform designed for high-throughput, fault-tolerant messaging.

Implements a publish-subscribe messaging model to enable real-time data processing at scale.

Ideal for constructing event-driven architectures and processing large data streams efficiently.

* *RabbitMQ:*

A robust message broker facilitating communication between different parts of a distributed system.

Implements the Advanced Message Queuing Protocol (AMQP) and supports message queuing, routing, and other messaging patterns.

* *JWT (JSON Web Token) Security:*

A standard for securing HTTP communication by generating tokens containing claims.

Facilitates authentication and authorization mechanisms in distributed systems, enhancing security and trust.

**Frontend:**

* *Vue.js:*

A contemporary JavaScript framework empowering developers to craft dynamic user interfaces.

Prioritizes simplicity and flexibility, facilitating seamless integration and scalability.

Implements reactive data binding for efficient UI updates and component reusability.

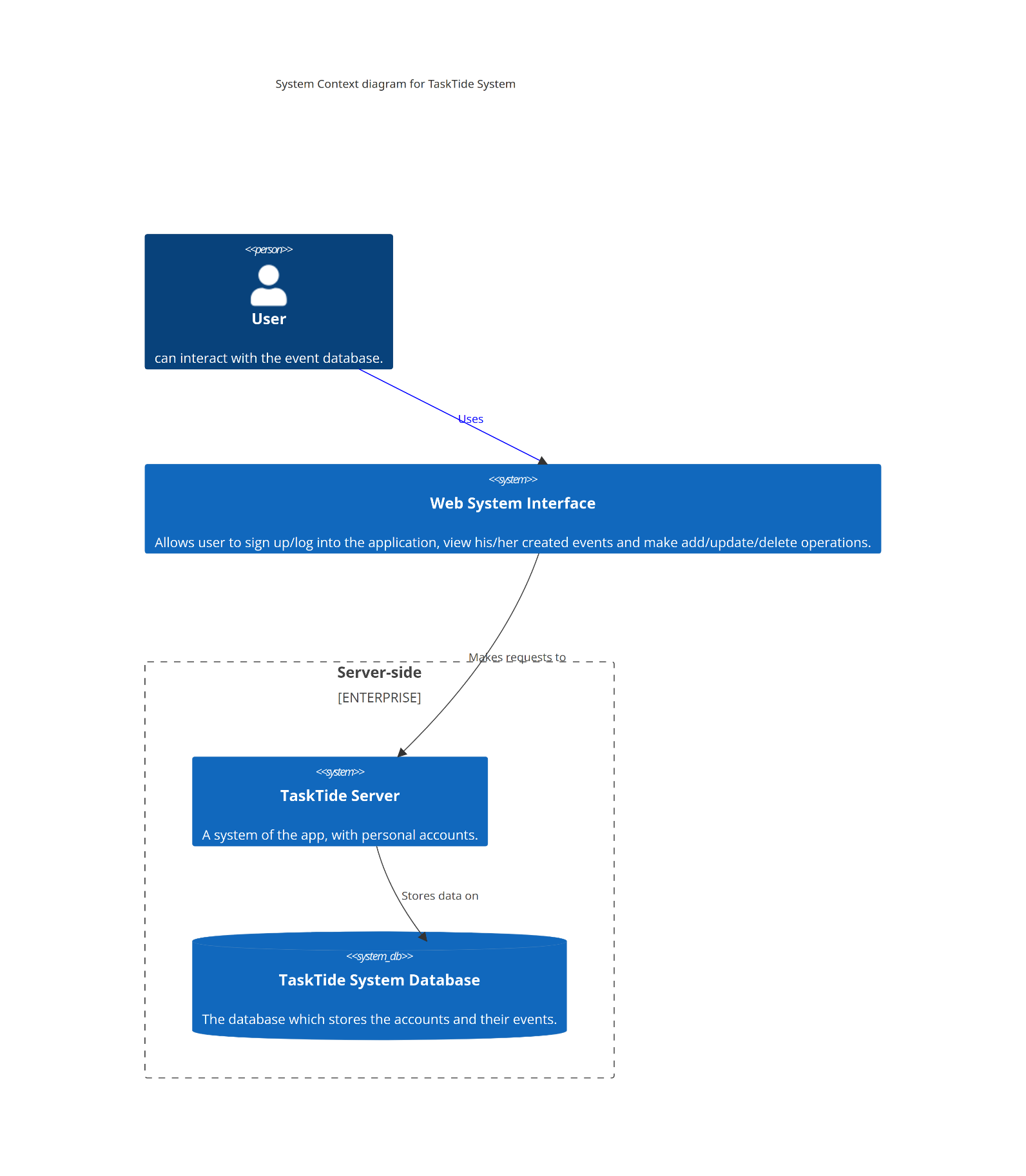
* *Module Federation:*

A feature within Webpack enabling dynamic loading of remote modules during runtime.

Supports the creation of micro-frontends, fostering independent development and deployment of application components.

**Overall:**

* *Integration:*
  + Docker containers ensure consistent deployment of the entire application stack, encompassing both backend and frontend components.
  + Spring Boot acts as the foundation for backend development, offering structured and modular development for server-side applications.
* *Communication:*
  + Kafka facilitates reliable communication between various microservices or components within the system, ensuring seamless event streaming.
  + RabbitMQ serves as the message broker, enabling efficient communication between different parts of the system using the Advanced Message Queuing Protocol (AMQP).
* *Data Storage:*
  + MySQL is chosen as the relational database, providing reliability and SQL support for structured data storage and retrieval.
* *Frontend Architecture:*
  + Vue.js empowers frontend development with its reactive and component-based architecture, facilitating modular and scalable user interface design.
  + Module Federation supports independent frontend module development and dynamic loading, enhancing scalability and maintainability.



A screenshot of a computer

Description automatically generated