PROJECT PROPOSAL

Transport Management System (TMS) for ITL Logistics Group

PART 1: Prioritized Sprint Backlog Item

The development of a Transport Management System (TMS) for ITL Logistics Group is currently at sprint 1 of the development timeline. In order to prioritize important tasks, our team has selected the following task from the Sprint Backlog:

Item name: Develop order and shipment tracking dashboard.

Item Description

The order and shipment tracking dashboard holds significant importance within the newly designed Transport Management System (TMS). It serves as a crucial functionality for the creation and management of orders, catering to both user needs and the tracking requirements of administrators and executives. Recognizing its significance, we have placed great emphasis on developing a user-friendly interface and have proactively deployed this feature on the Amazon Web Services (AWS) platform for thorough testing in the event of any potential errors. Furthermore, the backend logic development of this function has been prioritized to ensure the provision of essential information and insights to stakeholders. Additionally, our focus extends to ensuring future scalability and development possibilities for this feature.

Sub-tasks

- Design the user interface for the order creation and management page.
- Design order and shipment tracking dashboard.
- Implement server-side logic to handle order and shipment request.
- Set up the database to store order and shipment information.
- Implement real-time tracking of orders and shipments.
- Deploy currently developed functionalities on AWS.
- Perform System Integration Testing (SIT) on the orders and shipments page.

PART 2: Work Breakdown Structure

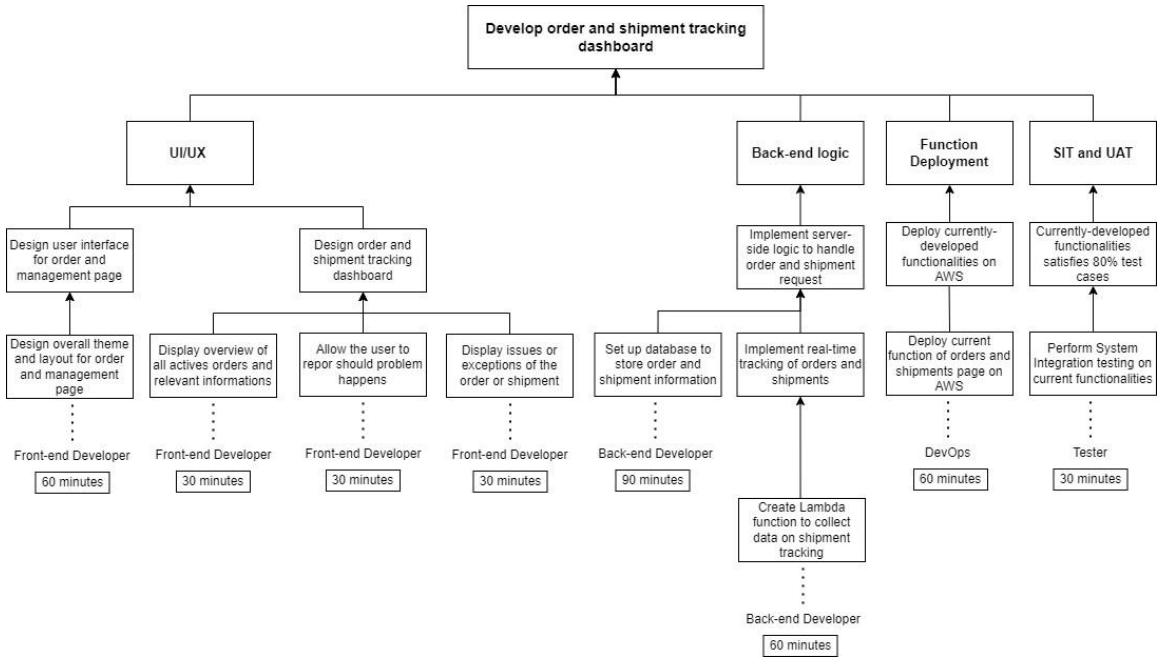

Figure 1: Work Breakdown Structure for order and shipment tracking dashboard.

Rationale:

The order and shipment tracking dashboard holds the utmost importance within sprint 1, necessitating the completion of multiple tasks across various sectors to ensure its desired functionality. Specifically, a total of 390 minutes has been allocated for this function, accounting for approximately 33% of the overall time allocation for sprint 1. The responsibility for completing these tasks lies with the front-end and back-end developers, DevOps, and the tester.

To ensure a user-friendly and comprehensive display of information for both users and administrators, we have allocated a substantial amount of time, specifically 150 minutes, for the design of an intuitive UI/UX for this functionality. Moreover, we have prioritized the backend development to enable a seamless and accurate presentation of information. This includes tasks such as data retrieval, processing, and rendering, all aimed at facilitating a smooth display experience.

Furthermore, the pre-deployment phase on AWS holds significant importance. During this phase, the DevOps and back-end developer will collaborate to develop and deploy the functionality on the AWS platform. This collaborative effort encompasses approximately 210 minutes, which accounts for roughly half of the allocated time for this item. This allows for thorough testing and validation of the function's performance and reliability.

Lastly, the tester will play a crucial role in ensuring the expected output by performing System Integration Testing on the current functionalities. This testing phase aims to validate the seamless integration of the order and shipment tracking dashboard with other existing system components, guaranteeing its smooth operation and accuracy.

PART 3: Quality Management

Quality management is crucial in the software development process, especially when adopting the Scrum framework. Scrum emphasizes iterative and incremental development, with frequent releases and feedback loops. In this dynamic environment, effective quality management ensures that the software meets the desired standards and satisfies customer expectations. Recognizing this importance, our team choose to adapt ISO 25010 framework for quality management.

Chosen characteristic: Reliability.

Characteristic and sub-characteristics details:

Among ISO 25010 characteristic, Reliability is a critical aspect. Reliability refers to the software's ability to maintain a specified level of performance under defined conditions for a specified period. It encompasses the software's ability to consistently perform its intended functions without failures or errors, as well as its ability to recover from failures gracefully.

Reliability is measured by various factors, including:

1. **Availability**: The software's readiness for use and its ability to be operational and accessible when needed, without excessive downtime or disruptions.
2. **Recoverability**: The software's ability to restore its normal functioning after a failure or disruption, including data recovery and system restoration mechanisms.
3. **Maturity**: The software's ability to provide information or services for the users when they make various request scenario.
4. **Fault tolerance**: The software's ability to continue functioning properly despite the presence of faults or errors in its components or environment.

The specific definition of done for these characteristics and sub-characteristics are as follows:

No.	Characteristics	Sub-Characteristics	Definition of Done criteria
1	Reliability	Availability	The feature should be accessible and responsive, downtime and unavailability must be significantly minimized to ensure 99.999% of availability.
2		Recoverability	At least 80% of the system's main functionalities must be able to operates normally after recovery from failure.
3		Maturity	The module functions must undergo SIT and UAT to operate reliably and consistently, and various scheduling scenarios must be tested to handle unexpected issues or failures.
4		Fault Tolerance	The feature's database needs to be replicated across a minimum of two regions. Additionally, scaling on demand should be implemented to handle any potential failures in the application server.

PART 4: Metric and Threshold for sub-characteristics

Sub-Characteristics	Metric	Threshold	Rationale
Availability	Uptime	99.999%	The feature should be operational and accessible for 99.999% of uptime period, experiencing minimal downtime.
Recoverability	Mean Time to Recover	< 1 hour	The average time take to restore the system to normal operation after a failure should be under 1 hour.
Maturity	Defect density	< 1.5 defects/KLOC	The number of defects identified per thousand lines of code (KLOC) should be under 1.5.
Fault Tolerance	Mean Time Between Failure	500 hours	The average time between two consecutive failures should be under 500 hours

Metrics and threshold explanation for the above sub-characteristics:

1. **Availability**: The metric for availability is percentage of uptime during a specific time period, in order to provide a high-availability service for the user, we define the threshold for this metric as 99.999%, meaning that the feature should be operational and accessible for 99.999% of uptime period.
2. **Recoverability**: The metric for recoverability is Mean Time to Recover (MTTR), which is the average time taken to restore the system to normal operation after a failure, we define the threshold for this metric as under 1 hour, meaning that the feature should be recovered to normal operation after a failure under 1 hour.
3. **Maturity**: The metric for maturity is defect density, which is the number of defects identified per thousand lines of code, we define the threshold for this metric as under 1.5 defects/KLOC, meaning that any errors, flaws, or issues present per thousand lines of code should be under 1.5 to meet the desired specifications.
4. **Fault Tolerance**: The metric for Fault Tolerance is Mean Time Between Failure (MTBF), which is the average time between two consecutive failures, we define the threshold for this metric as 500 hours, meaning that the feature should operate without failure for at least 500 hours before encountering the next failure.

By adhering to these quality management practices and the specified metrics and thresholds, we can ensure that our feature meets the desired specifications and delivers a reliable, robust, and high-quality experience to the users.