



# Quality Management

**Company:** Top Deal Auto Melbourne, Australia

**Software:** Car Selling Website

**Team name:** Prestige K/DA

**Team Members:**

Name	ID	Roles
Pham Duc Linh	103792371	Product Owner - BA
Pham Anh Vu	103806447	Solutions, IT Architect
Nguyen Thanh Dat	103804881	Project Manager - Scrum Master
Tran Tuan Nam	103792643	Lead Developer (BE)
Phung Xuan Tung	103792054	Lead Developer (FE)

**Tutorial class:** Fri 1:00 PM DT7.2

**Tutor’s Name:** Dr. Pham Thi Kim Dzung

## **Definition of Done**

### **Functionality Suitability**

#### **Functional Correctness**

##### **Condition**

- User purchasing product flow works as per describe without fault (user add product to cart, user gets to checkout, user insert payment information, transaction success, sales record gets saved, appropriate amounts gets deducted, email transaction information to end user). If a user refunds, the transaction must be reverted and logged.
- Administrators' product flow works without fault (admin logs in, admin adds/delete/edit product information, changes are reflected to end users, changes are saved and reversible). Data export must allow .csv, .xlsx formats. Separate process node must be created specifically for data extraction purposes. Extract operations must be run from the analytics database, not the main database.

##### **Quality Metric**

- All end user UI, UX, Functional tests must allow only 5% failure. All administrator UI, UX, functional tests must allow only 5% failure rate, per industry standard

##### **Measurement**

- Manual UI, UX, Workflow testing by QA, QC team.

### **Functional Appropriateness**

##### **Condition**

- Allow users to perform buying transactions, account creation
- Allow administrators to manage products (viewing, adding, removing, editing) and users (viewing, adding, removing, editing)

##### **Quality Metric**

- Able to complete the core existing operations of Top Deal Auto's existing system
- Able to match 95% of other additional functionalities

##### **Measurement**

- Performing existing workflow on new system.

# Performance Efficiency

## Time Behaviour

### Condition

- Low latency of content delivery to users globally. Administrator tasks completed with minimal delay. Transactions completed, reflected with low latency

### Quality Metric

- **Content Delivery:** 10ms response time maximum for core regions (Oceania, Asia Pacific), 50ms response time maximum for other regions (East Asia, Europe, US)
- **Transactional Speed:** All transactions must be updated, reflected on the main database within maximum 10 ms. Content must be synced in real time (maximum delay 10ms).
- **Administrator tasks:** Insert, delete, update operations must take maximum 10ms. Read operation takes a maximum of 5ms.

### Measurement

- Lambda Speed Benchmarks on client side, AWS CloudWatch on server, network side

## Resource Utilization

### Condition

- No single server allowed to exceed the required functional resources. Automatically scale and notify admins when such cases occur.
- Network traffic in a region must be distributed evenly between servers in the same region.
- Each region must have a master database for both Read/Write operations, a read replica for only read operations. More traffic must be routed to the Read Replica.

### Quality Metric

- **Server:** 70% CPU, 80% RAM Utilization
- **Network:** Even distribution between all servers in the same region
- **Database:** 70% read request to Read Replica, 30% read request to main database, 100% write request to main database.

### Measurement

- Scripted test events with AWS Lambda, notifications with CloudWatch, SNS

## Capacity

### Condition

- Enough server and network scales to match resource utilization while minimizing cost

## Quality Metric

- **Server:** Minimum 1 per region, Maximum 5 per region
- **Database:** Minimum 1 main database, 1 read replica per region. Maximum 1 main database, 5 read replica per region
- **Block store:** Move items from S3 basic to S3 Glacier after 60 days of no access

## Measurement

- AWS Advisor
- AWS configuration
- AWS Health check

# Usability

## User interface aesthetics

### Condition

- Visually appealing design consistent with brand, adhering to modern standards.

### Metrics

- User satisfaction rating for visual appeal ( $\geq 4$  out of 5), adherence to design standards (compliance rate  $\geq 90\%$ ).

### Measurement

- User surveys and expert design standard reviews.

## User error protection

### Condition

- The website prevents user errors and provides clear recovery options.

### Metrics

- Number of user errors encountered (reduction by  $\geq 50\%$  from initial tests), user ratings for error messages' clarity and helpfulness ( $\geq 4$  out of 5).

### Measurement

- Tracking user errors during testing and collecting user feedback.

# Reliability

## Fault Tolerance

### Condition

- The website remains operational and minimizes data loss in the event of errors or failures.

### Metrics

- **Error Impact Severity:** Less than 10% of errors should significantly impact user experience.
- **System Downtime:** System should have an uptime of 99.5% or higher.
- **Data Loss Instances:** Data loss should occur in less than 0.1% of all system errors.

### Measurement

- **Error Impact:** Monitor and categorize the impact of errors through error logging and user feedback.
- **System Downtime:** Use server monitoring tools to track uptime and downtime.
- **Data Loss Instances:** Implement logging mechanisms to track any instances of data loss and measure against the total number of errors.

## Recoverability

### Condition

- The website can recover quickly and completely after a failure, restoring all necessary data and functionality.

### Metrics

- **Recovery Time Objective (RTO):** The system should be able to recover from any failure within 2 hours.
- **Recovery Point Objective (RPO):** Data should be recoverable to a state no more than 30 minutes prior to the failure.
- **Recovery Success Rate:** At least 98% of recovery attempts should be successful.

### Measurement

- **Recovery Time:** Measure the actual recovery time in response to different types of failures during testing.
- **Data Recovery Point:** Use data backup and restore tests to ensure that data can be recovered to the desired point.
- **Recovery Success:** Track and analyze the success rate of recovery processes during testing and real incidents.

# Security

## Confidentiality

Ensures that information is accessible only to those authorized to have access.

### Condition

- User data and sensitive information are protected from unauthorized access.

### Metrics

- **Incidents of unauthorized data access:** Zero incidents.
- **Encryption strength:** Use of industry-standard encryption protocols (e.g., TLS 1.3 for data in transit).
- **Access control compliance rate:**  $\geq 98\%$  compliance with access control policies.

### Measurement

- **Security Audits and Penetration Testing:** Regular audits and penetration tests to check for vulnerabilities.
- **Encryption Verification:** Regular checks to ensure encryption protocols are up-to-date.
- **Access Control Monitoring:** Continuous monitoring and logging of access to sensitive data, with periodic reviews.

## Integrity

### Condition

- Data is protected against unauthorized modifications, ensuring its accuracy and reliability.

### Metrics

- **Data alteration incidents:** Zero unauthorized alterations.
- **Checksum or Hash Verification Success Rate:**  $\geq 99\%$  success rate in verifying data integrity.
- **System audit log accuracy:** 100% accuracy in recording data transactions.

### Measurement

- **Integrity Checks:** Regularly scheduled integrity checks using checksums or cryptographic hashes.
- **System Audit Logs:** Implement and monitor system audit logs to track changes and access to data.
- **Incident Reporting and Analysis:** Immediate investigation and reporting of any integrity-related incidents.

# Portability

## Adaptability

### Condition

- The website is fully functional and responsive on a wide range of popular web browsers, operating systems, and electronic devices.

### Metrics

- **Operating Systems:** Demonstrates seamless functionality and maintains performance standards on Windows, Linux, and MacOS.
- **Electronic Devices:** Achieves optimal responsiveness and maintains performance on Desktops, Laptops, Smartphones, and Tablets.

### Measurement

- Conduct comprehensive testing across multiple environments, including different combinations of operating systems, web browsers, and device types.
- Performance and responsiveness are evaluated through a series of practical tests to ensure compatibility and efficiency in each environment.

## Team Discussion

Member	Decision Explanation
Pham Duc Linh	I fully support the team's Definition of Done (DoD) document for our project. The criteria outlined are thorough and provide a clear framework for assessing the quality and completeness of our deliverables. This document will be a valuable guideline for our future development efforts, ensuring consistency and reliability in our work.
Nguyen Thanh Dat	<p>I'm completely on board with our team's Definition of Done (DoD) for the project. It's really impressive how detailed and thought-out the criteria are. They lay down a solid foundation for evaluating our work, making sure that everything we deliver is up to the high standards we're aiming for.</p> <p>This DoD isn't just a set of guidelines; it's like a roadmap that will steer us in the right direction throughout the development process. It's reassuring to know that we have such a comprehensive framework in place. It'll help us stay on track, maintain consistency, and ensure that every aspect of our project is of top-notch quality.</p> <p>I'm really excited to see how this DoD will positively impact our workflow and final outputs. Great job, team! Let's keep this momentum going and create something we're all proud of.</p>
Pham Anh Vu	I have some skepticisms when it comes to the definition of done that

	<p>we ended up with, especially when it comes to the capacity and utilization metrics. The utilization metrics in my opinion needs to be higher, as the current level, when scaled globally under high usage events (i.e. Sales) can exponentially increase the number of resources, and in turn, the price. I think increasing the utilization and reducing the performance requirements will be more financially beneficial in the long run.</p> <p>Another point to add, having 1 server per region is a bit overkill, as a region's server(s) is fully capable of handling load from another region with minimal performance latency. And the case of multiple regions failing at the same time is while technically possible, not really probable, so accounting for this is unnecessary.</p>
<b>Tran Tuan Nam</b>	<p>Overall I am on board with the team's DoD for this project. The document has defined and elaborate in detail every aspect for the completion of this project. I strongly believe that should the final result satisfy all of the criteria mentioned in this document, our work should be able to solve all of the customer's requirements in good quality. However, the metric for resource utilization should be a bit more strict since I can tell that the server can still run efficiently with less resources that what was stated in this document.</p>
<b>Phung Xuan Tung</b>	<p>I agree with the Definition of Done for this project, as it covers the essential aspects of functionality, performance, resource utilization, and usability. The conditions, metrics, and measurements are clear and reasonable, and they align with the project goals and user expectations. However, I think the capacity could be improved in some areas. The minimum and maximum number of servers per region could be adjusted based on the expected traffic and demand. For example, if the traffic is low in a region, one server might be enough, but if the traffic is high, five servers might not be sufficient. A dynamic scaling strategy could be implemented to optimize the server capacity and cost. Also, the ratio of read and write requests to the main database and the read replica could be fine-tuned based on the actual usage patterns. For example, if the read requests are more frequent than the write requests, the read replica could handle more than 70% of the read requests, reducing the load on the main database. A load balancing mechanism could be applied to distribute the requests evenly among the available databases.</p>
<b>Team Final Agreement</b>	
<ul style="list-style-type: none"> <li>- The team broadly agrees on the Definition of Done (DoD), valuing its thoroughness and comprehensive framework.</li> <li>- However, there are concerns about capacity, utilization metrics, and server distribution, suggesting possible revisions for efficiency and cost-effectiveness.</li> <li>- Despite these concerns, the overall sentiment towards the DoD is positive, with a willingness to adapt based on practical considerations.</li> </ul>	