Performance Testing Phase

Date	27 June 2025
Team ID	LTVIP2025TMID31086
Project Name	calculating family expenses using service now
Maximum Marks	4 Marks

The performance testing phase served as a vital checkpoint in validating the system's ability to deliver consistent, real-time financial management support under various usage conditions. Given the nature of household expense tracking—often involving frequent updates, multi-user input, and data dependencies—ensuring the platform's responsiveness, accuracy, and scalability was paramount.

While functionality testing ensures a system "works," performance testing confirms that it works efficiently and reliably as demands grow. The testing focused on ensuring seamless interaction between the core data tables, maintaining data integrity, and supporting real-time user experiences under both normal and high-load conditions.

Objectives of Performance Testing

The key goals for this phase were to:

• Ensure that automated business logic triggers executed accurately under real-time and high-volume conditions.

• Maintain fast load times and responsive UI behavior regardless of dataset size.

• Validate that table relationships and calculated fields remained accurate and synchronized.

• Simulate real-world usage scenarios, including multi-user inputs and varying data volumes.

• Identify any bottlenecks that could affect user adoption or system reliability over time.

Testing Scope and Activities

1. Business Rule Validation

Business rules were central to maintaining dynamic updates between the Daily Expenses and Family Expenses tables. During testing, a variety of scenarios—such as adding, editing, and deleting expense entries—were executed to verify that the related aggregation logic in the Family Expenses table responded in real time.

Results:
The custom business rules fired consistently and correctly, updating cumulative totals per family member without delay. Edge cases, like deleting entries or modifying historical records, were handled accurately, with no data loss or lag.

2. User Interface Responsiveness

User experience was assessed through responsiveness testing, focusing on form load times, field refresh rates, and the update latency of dynamically calculated fields.

Results:
Forms rendered quickly (within 1–2 seconds) even after configuring advanced field logic and client scripts. Dynamically populated fields such as total expenses, category breakdowns, and custom notifications updated near-instantly, ensuring a smooth user interaction.

3. Data Integrity Testing

This involved stress-testing the data relationships by inputting random, high-frequency updates in both tables, out of chronological order, and with intentional gaps or edge-case values (e.g., large amounts, missing categories).

Results:
Data linkages remained intact. No orphan records, duplicate entries, or calculation errors were found. Record synchronization between tables remained strong, demonstrating resilience to disorderly data inputs.

4. Field Behavior Checks

Default field values, mandatory requirements, and read-only field protections were thoroughly tested for reliability across multiple forms and entry states.

Results:
All field behaviors operated as expected. Default values populated based on current date and user profile, mandatory fields blocked record submission when incomplete, and protected fields were successfully enforced across all user roles.

5. Scalability and Load Testing

Simulated datasets with hundreds of entries across multiple users and extended date ranges were used to test how the system handled growing volumes of data and concurrent input activity.

Results:
The system handled data growth gracefully. Performance degradation was negligible even with 500+ records logged, and dashboard charts adapted to the scale without error or rendering delay. ServiceNow's platform capabilities effectively supported horizontal scaling, reinforcing confidence in the solution's longevity.

Conclusion

The performance testing phase validated that the household financial management system was not only functional but also resilient, scalable, and user-friendly under real-world conditions. By simulating various usage scenarios—from casual daily entry to high-volume, multi-user engagement—the team ensured that users could rely on the system without facing delays, errors, or data inconsistencies.

This phase helped highlight the robustness of ServiceNow as a backend and justified its use beyond ITSM. With successful performance testing complete, the system is well-prepared for pilot deployment and user feedback collection, ensuring its readiness for broader adoption.