| | |
|---|---|
| **Date** | **02 November 2025** |
| **Team ID** | **NM2025TMID05269** |
| **Project Name** | **Calculating Family Expenses** |
| **Maximum Marks** | **5 Marks** |

# Project Planning Phase

## Product Backlog, Sprint Schedule, and Estimation (4 Marks)

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Update Set Configuration | USN-1 | As a developer, I can create a new update set to capture and migrate all configuration changes for the Family Expense Tracker project. | 2 | High | Nabeel Ahmed N |
| Sprint-1 | Table Creation | USN-2 | As an admin, I can create a Family Expenses table to store and manage cumulative expense data for each family. | 3 | High | Benit K |
| Sprint-2 | Table Creation | USN-3 | As an admin, I can create a Daily Expenses table to record individual expense entries and associate them with family data. | 3 | High | Lekshmanan M |
| Sprint-2 | Relationship Setup | USN-4 | As a system user, I can establish a relationship between Family Expenses and Daily Expenses tables for data linkage. | 3 | Medium | Naveen MS |
| Sprint-3 | UI Configuration | USN-5 | As a system user, I can configure a Related List on Family Expenses to view all associated Daily Expense records. | 3 | High | Nabeel Ahmed N |
| Sprint-3 | Automation Rule | USN-6 | As a developer, I can create a Business Rule that updates total expenses automatically when a new Daily Expense is added or modified. | 3 | High | Benit K |
| Sprint-4 | Relationship Configuration | USN-7 | As a developer, I can configure advanced relationship logic using scripts to refine data association between tables. | 3 | Medium | Lekshmanan M |
| Sprint-4 | Documentation | USN-8 | As a developer, I can prepare final documentation for submission, including implementation steps and validation results. | 2 | Medium | Naveen MS |

## Project Tracker, Velocity & Burndown Chart: (4 Marks)

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed | Sprint Release Date |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 4 Days | 18 October 2025 | 21 October 2025 | 20 | 21 October 2025 |

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed | Sprint Release Date |
|---|---|---|---|---|---|---|
| Sprint-2 | 20 | 4 Days | 22 October 2025 | 25 October 2025 | 20 | 25 October 2025 |
| Sprint-3 | 20 | 4 Days | 26 October 2025 | 29 October 2025 | 20 | 29 October 2025 |
| Sprint-4 | 20 | 4 Days | 30 October 2025 | 02 November 2025 | 20 | 02 November 2025 |

Average Velocity = (Total Story Points Completed) / (Total Duration in Days)
Total: 80 Story Points over 16 Days → Velocity = 5 Points/Day

A Burndown Chart is a graphical representation of the remaining work versus time for each sprint. It helps track sprint progress and visualize the completion rate of tasks over the sprint timeline. In this project, the burndown chart shows consistent progress across all four sprints, demonstrating steady task completion and sprint reliability.

## References


• ServiceNow – https://www.servicenow.com/products/platform.html

• ServiceNow Tables and Relationships – https://docs.servicenow.com/bundle/tokyo-platform-administration/page/administer/table-administration/concept/table-relationships.html

• ServiceNow Business Rules – https://docs.servicenow.com/bundle/tokyo-application-development/page/script/business-rules/concept/business-rules.html

• ServiceNow Update Sets – https://docs.servicenow.com/bundle/tokyo-application-development/page/build/system-update-sets/concept/update-sets.html