WolfCafe

CSE 510 Software Engineering : Project 1b1

Contributors : Janam Patel, Namit Patel, Pranshav Patel, Vivek Vanera

Question 1 : 30 New use cases generated by RAG.
Response :

1. **Log In**
   Preconditions : User has an active account. System is operational and accessible.
   Main Flow :
   (a) User navigates to the MFOS login page.
   (b) System displays login form (username, password).
   (c) User enters username and password.
   (d) User clicks "Log In".
   (e) System validates credentials.
   (f) System grants access and redirects to dashboard.
   Subflows : None.
   Alternative Flows :
   A1 : Invalid Credentials – System shows "Invalid username or password."
   A2 : Account Locked – System shows "Your account is locked."
   A3 : Forgot Password – User clicks reset link to recover credentials.
2. **Log Out**
   Preconditions : User is logged in.
   Main Flow :
   (a) User clicks "Log Out".
   (b) System terminates session.
   (c) System redirects to login page.
   Alternative Flows :
   A1 : Session Timeout – Auto logout after inactivity.
3. **Manage User Profile**
   Preconditions : User is logged in.
   Main Flow :
   (a) User navigates to "My Profile".
   (b) System displays editable fields.
   (c) User updates profile info.
   (d) User clicks "Save Changes".
   (e) System validates and updates record.
   Subflows : Validate Input.
   Alternative Flows :
   A1 : Invalid Input – System highlights errors.
   A2 : Cancel Changes – Discards updates.
4. **Search System Data**
   Preconditions : User is logged in.
   Main Flow :

(a) User opens search bar.
(b) System displays input and filters.
(c) User enters query and submits.
(d) System executes search and shows results.

Alternative Flows :
A1 : No Results Found – System shows "No results found."
A2 : Refine Search – User adjusts filters.

5. **Generate System Report**
Preconditions : User has reporting permissions and data exists.
Main Flow :
(a) User navigates to "Reports".
(b) System shows available reports and options.
(c) User selects type and configures parameters.
(d) User clicks "Generate Report".
(e) System processes and generates output.
(f) System displays download link.

Subflows : Data Retrieval and Processing.
Alternative Flows :
A1 : Insufficient Data – No data available.
A2 : Report Failure – Error message displayed.

6. **Create New Project**
Preconditions : User has Project Manager/Admin role.
Main Flow :
(a) User selects "Create New Project".
(b) System shows project form.
(c) User enters details and saves.
(d) System validates and creates project.

Alternative Flows :
A1 : Invalid Input – Errors highlighted.

7. **Define Project Charter**
Preconditions : Project exists in Planning phase.
Main Flow :
(a) User selects "Project Charter".
(b) System shows charter fields.
(c) User fills details and saves.
(d) System validates and stores charter.

Alternative Flows :
A1 : No Permission – Access denied.

8. **Prepare Project Documentation**
Preconditions : Project exists.
Main Flow :
(a) User goes to "Documentation".
(b) User uploads file with metadata.
(c) System validates and stores file.

Alternative Flows :

A1 : Upload Failed – Retry needed.
A2 : View/Download existing docs.

9. **Manage Project Budget**
   Preconditions : Project exists. User has finance rights.
   Main Flow :
   (a) User goes to "Budget" section.
   (b) Adds/adjusts items.
   (c) System recalculates totals.

   Alternative Flows :
   A1 : Budget Exceeded – Warning shown.

10. **Allocate Project Resources**
    Preconditions : Project exists with available resources.
    Main Flow :
    (a) User selects "Allocate Resource".
    (b) System shows allocation form.
    (c) User assigns resource.
    (d) System validates and updates schedule.

    Alternative Flows :
    A1 : Resource Conflict – Suggest alternatives.
    A2 : Deallocate Resource – Removes assignment.

11. **Create Project Plan**
    Preconditions : Project exists.
    Main Flow :
    (a) User goes to "Planning".
    (b) Adds tasks with dependencies.
    (c) System validates and updates Gantt chart.

    Alternative Flows :
    A1 : Invalid Dependency – Error shown.
    A2 : Edit Task – Modify existing task.

12. **Submit Project Plan for Approval**
    Preconditions : Project plan complete.
    Main Flow :
    (a) User submits plan.
    (b) System validates and sets status to Pending Approval.
    (c) Notifications sent to approvers.

    Alternative Flows :
    A1 : Plan Incomplete – Submission blocked.

13. **Manage Procurement Request**
    Preconditions : Project requires hardware/software.
    Main Flow :
    (a) User creates procurement request.
    (b) System validates and routes for approval.

    Alternative Flows :
    A1 : Request Rejected – Notified with comments.

14. **Track Design Task Progress**
    Preconditions : Design tasks exist.
    Main Flow :
    (a) User selects design task.
    (b) Updates progress and attaches artifacts.
    (c) System saves updates.
    Alternative Flows :
    A1 : Task Blocked – System prompts reason.
15. **Record Testing Results**
    Preconditions : Project in Testing phase.
    Main Flow :
    (a) User selects test case.
    (b) Enters results, status, and evidence.
    (c) System saves results.
    Alternative Flows :
    A1 : Fail Case – Prompts defect creation.
16. **Initiate Project Audit**
    Preconditions : Project is active.
    Main Flow :
    (a) User schedules audit.
    (b) System validates and notifies auditors.
    Alternative Flows :
    A1 : Generate Audit Report post-completion.
17. **Update Project Records**
    Preconditions : Project active.
    Main Flow :
    (a) User updates records.
    (b) System saves changes.
    Alternative Flows :
    A1 : Delete Record – Archive entry.
18. **Close Project**
    Preconditions : Deliverables complete.
    Main Flow :
    (a) User initiates closure.
    (b) System verifies completion and archives.
    Alternative Flows :
    A1 : Closure Blocked – Outstanding items remain.
19. **Monitor Task Dependencies**
    Preconditions : Project plan exists.
    Main Flow :
    (a) User views dependencies in Gantt chart.
    (b) System highlights delays.
    Alternative Flows :
    A1 : Dependency Alert – Notifications sent.

20. **View Project Timeline**
Preconditions : Defined plan exists.
Main Flow :
(a) User opens "Timeline" view.
(b) System shows Gantt chart with tasks and dependencies.
Alternative Flows :
A1 : Apply Filters.
A2 : Highlight Critical Path.

21. **Register New DFDO Facility**
Preconditions : User has admin rights.
Main Flow :
(a) User registers facility.
(b) System validates and stores record.
Alternative Flows :
A1 : Duplicate Facility – Warning issued.

22. **Update DFDO Facility Profile**
Preconditions : Facility registered.
Main Flow :
(a) User edits facility profile.
(b) System saves updates.
Alternative Flows :
A1 : Regulatory Notification required.

23. **Record Food Handling Practices**
Preconditions : Facility registered.
Main Flow :
(a) User documents food handling practices.
(b) System validates and reassesses sink requirements.
Alternative Flows :
A1 : Inconsistent Data – Prompts clarification.

24. **Review Sink Requirement Compliance**
Preconditions : Practices recorded.
Main Flow :
(a) User reviews required vs actual sinks.
(b) System highlights discrepancies.
Alternative Flows :
A1 : Update Sink Configuration.

25. **Document Handwashing Sink Details**
Preconditions : Facility registered.
Main Flow :
(a) User adds handwashing sink details.
(b) System validates and updates record.
Alternative Flows :
A1 : Missing Sink – Warning shown.

26. **Document Food Preparation Sink Details**
Preconditions : Facility registered.
Main Flow :

(a) User documents food preparation sink.

(b) System validates and updates record.

Alternative Flows :

A1 : Shared Sink – Non-compliance warning.

A2 : Alternative Procedure required.

27. **Document Alternative Produce Washing Method**
Preconditions : Facility lacks prep sink.
Main Flow :

(a) User documents alternative washing procedure.

(b) System saves and updates compliance.

Alternative Flows :

A1 : Procedure Rejected by officer.

28. **Record Restroom Facility Compliance**
Preconditions : Facility registered.
Main Flow :

(a) User records restroom compliance.

(b) System saves and calculates status.

Alternative Flows :

A1 : Non-compliance – Requires corrective plan.

29. **Record Backflow Prevention Status**
Preconditions : Facility registered.
Main Flow :

(a) User documents backflow devices.

(b) System validates and schedules alerts.

Alternative Flows :

A1 : Device Failure – Corrective action required.

30. **Generate Compliance Summary**
Preconditions : Facility compliance data recorded.
Main Flow :

(a) User generates compliance report.

(b) System aggregates data and outputs report.

Alternative Flows :

A1 : Export Data – CSV format.

A2 : Drill Down on non-compliance.

Question 2 : Comparison of responses generated by Grok and Gemini-based RAGs.
Response : The Gemini response provided a solid baseline with 30 use cases, focusing mainly on MFOS as a project management and compliance system. Its strengths lie in clear structure (preconditions, main flow, subflows, and alternative flows) and a comprehensive breakdown of each activity, such as audits, budget management, and DFDO compliance. However, Gemini's style leans toward a formal academic tone, sometimes verbose, and occasionally repetitive in alternative flows. It also restricted itself to the MFOS domain, which, while precise, limited coverage of generalizable scenarios.

The Grok response, by contrast, feels more practical, domain-diverse, and user-centric. It not only covers 30 use cases but also expands into e-commerce-like flows (orders, payments, loyalty, promo codes, receipts), along with administrative and accessibility scenarios. The format is highly consistent : each use case follows a templated flow with tags for substeps (e.g., [Access Orders], [Manage User]), improving readability and

modularity. Grok also introduces more robust alternative flows, covering realistic edge cases like network failures, invalid inputs, outdated standards, or supplier integration conflicts.

From a content quality perspective, Gemini emphasizes depth in fewer domains, while Grok emphasizes breadth and completeness across roles. In terms of inference time, Grok's concise subflow notation likely makes it easier to parse programmatically (e.g., for automated RAG evaluation or testing), whereas Gemini's longer narrative steps might take longer to process. For clarity and consistency, Grok clearly outperforms by avoiding redundancy and ensuring uniform flow across all use cases.

In summary, Gemini delivers a research-style, MFOS-specific set of use cases, whereas Grok offers a production-ready, multi-role, system-wide library of use cases. Grok's improvements in readability, variety, and edge case handling make it better suited for applied system design and implementation, while Gemini remains valuable for academic rigor and deep domain alignment.

Question 3 : Report the total cost of LLM usage.
Response : $0. Both LLMs (Grok and Gemini) provide free APIs.