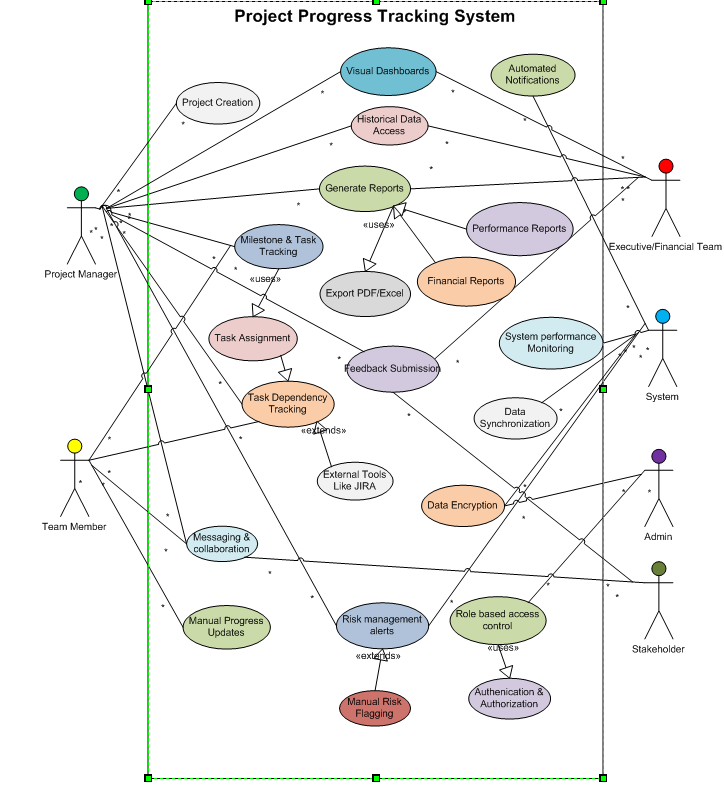
**WATERFALL MODEL DOCUMENTS PART – 2**

**Project Title:** Project Progress Tracking Feature in Sparsh Application

**Document 6 - Please prepare a use case diagram, activity diagram and a use case specification document.**

**Use Case Diagram**



**Use Case Specification Document**

|  |  |
| --- | --- |
| **Use Case Spec** | **Project Creation** |
| **Description** | Allows project managers to create and initiate new projects |
| **Actors** | Project Managers, System |
| **Pre-conditions** | User must be logged in and have required Permissions |
| **Post Conditions** | Project created successfully and visible on dashboard |
| **Basic Flows** | 1. Project Manager login in 2. Navigates to “Create new Project” 3. Enter Project details and assign Team Members 4. Saves project and system confirms creation |
| **Alternate Flows** | If user cancels before submit, then project is not created |
| **Exceptional Flows** | If mandatory fields are missing, system shows an error |
| **Assumptions** | Project manager has clarity on project requirements |
| **Constraints** | Project names must be unique |
| **Dependencies** | Requires database access to store project details |
| **Input** | Project name, description, start & end date, assign team members |
| **Output** | Confirmation messages, project created in the system |
| **Business rules** | Only authorized users can create projects |
| **Miscellaneous Info’s** | Project Cannot be deleted after creation |

|  |  |
| --- | --- |
| **Use Case Spec** | **Milestone & Task Tracking** |
| **Description** | Allows tracking of project milestones and tasks |
| **Actors** | Project Managers, Team Member ,System |
| **Pre-conditions** | Project must be created and tasks assigned |
| **Post Conditions** | Updates project status based on completed milestones |
| **Basic Flows** | 1. PM or TM views milestone progress & Updates task status 2. System auto updates milestone completion percentage |
| **Alternate Flows** | If task dependencies exist, updates must follow sequence |
| **Exceptional Flows** | If task is deleted, system prompts for confirmation |
| **Assumptions** | Tasks are updated regularly |
| **Constraints** | Task tracking is only available within active projects |
| **Dependencies** | Requires database updates and role based access |
| **Input** | Task updates, completion status |
| **Output** | Updated progress dashboard |
| **Business rules** | Tasks must be assigned before they can be tracked |
| **Miscellaneous Info’s** | Project history is maintained for audit purposes |

|  |  |
| --- | --- |
| **Use Case Spec** | **Manual Progress Updates** |
| **Description** | Allows teams to manually updates their progress |
| **Actors** | Project Managers, Team member, System |
| **Pre-conditions** | Task must be assigned to the team member |
| **Post Conditions** | Task completion status is updated in the system |
| **Basic Flows** | 1. Team member logs into the system 2. Selects the assigned task and input progress details 3. Submit the update 4. System records and reflects the update in reports |
| **Alternate Flows** | If tasks is already completed, updates are restricted |
| **Exceptional Flows** | If incorrect data is entered, an error message is shown |
| **Assumptions** | Team members provide accurate progress updates |
| **Constraints** | Only one update is allowed per task per day |
| **Dependencies** | Requires project and task assignment records |
| **Input** | Task ID, Updated Status and comments |
| **Output** | Updated task list |
| **Business rules** | Only assigned user can update tasks |
| **Miscellaneous Info’s** | Task updates are logged for audit purposes |

|  |  |
| --- | --- |
| **Use Case Spec** | **Visual Dashboard** |
| **Description** | Displays real time project status in graphical format |
| **Actors** | Project Managers, Upper management |
| **Pre-conditions** | Projects must have assigned tasks and milestones |
| **Post Conditions** | Displays updated progress metrics |
| **Basic Flows** | 1. User logs into the system 2. Navigates to the dashboard 3. Select project to view 4. System displays charts and graphs representing project progress |
| **Alternate Flows** | User can apply filters for specific views |
| **Exceptional Flows** | If project data is unavailable an error message is shown |
| **Assumptions** | Data is updated in real time |
| **Constraints** | Dashboard refresh every 5 minutes |
| **Dependencies** | Requires real time data access |
| **Input** | Project ID, filters |
| **Output** | Graphical reports, charts |
| **Business rules** | Users can only view projects they are assigned |
| **Miscellaneous Info’s** | Export options like PDF and excel are available |

|  |  |
| --- | --- |
| **Use Case Spec** | **Generate Reports** |
| **Description** | Allows users to generate projects status reports |
| **Actors** | Project Managers, Upper management |
| **Pre-conditions** | Project must have recorded progress data |
| **Post Conditions** | System generates and exports reports |
| **Basic Flows** | 1. User navigates to reports sections 2. Select reports type 3. System generates the report 4. User downloads or shares the report to officials |
| **Alternate Flows** | User applies filters to generate a specific report |
| **Exceptional Flows** | If no data exists, system shows an empty report message |
| **Assumptions** | Reports are generate regularly |
| **Constraints** | Report generation time depends on data volume |
| **Dependencies** | Requires access to historical project data |
| **Input** | Project ID, report Type |
| **Output** | Report file like PDF, Excel, CSV |
| **Business rules** | Only authorized users can generate projects |
| **Miscellaneous Info’s** | Reports can be auto scheduled |

|  |  |
| --- | --- |
| **Use Case Spec** | **Task Assignment and Dependencies** |
| **Description** | Allows project managers to assign task and set dependencies between tasks |
| **Actors** | Project Managers, Team member |
| **Pre-conditions** | Project must be created and active |
| **Post Conditions** | Tasks are assigned, dependencies are updated in the system |
| **Basic Flows** | 1. Project manager select a project 2. Assigns tasks to team members 3. Defines dependencies between tasks 4. Saves changes and the system updates assignments |
| **Alternate Flows** | If a task is reassigned , notification are sent to affected users |
| **Exceptional Flows** | Without proper authorization, error message is shown to user |
| **Assumptions** | A tasks are clearly defined before assignment to user |
| **Constraints** | A tasks cannot start unless its dependencies are met |
| **Dependencies** | Requires project creation and team assignment |
| **Input** | Task details, assigned users, dependency structures |
| **Output** | Updated task list, notification sent |
| **Business rules** | Only project manager can assign or modify task dependencies |
| **Miscellaneous Info’s** | Dependencies should be clearly visualized in the system |

|  |  |
| --- | --- |
| **Use Case Spec** | **Automated Notifications & Alerts** |
| **Description** | Allows system to send notification to users for task completions, updated task |
| **Actors** | Project Managers, Team member, System |
| **Pre-conditions** | Tasks or milestone must be configured for tracking |
| **Post Conditions** | Notifications must be sent to appropriate users |
| **Basic Flows** | 1. System monitors project status 2. Task completion triggers notifications 3. Delays trigger alerts to project stakeholders |
| **Alternate Flows** | 1. User manually configures notification settings 2. Notifications preference override system defaults |
| **Exceptional Flows** | 1. System fails to send notification due to network issue 2. Logs errors and retries notifications |
| **Assumptions** | Users will have access to email/SMS/app notifications |
| **Constraints** | Limited to predefined event triggers |
| **Dependencies** | Requires active internet and user registration |
| **Input** | Task completion data, risk assessments |
| **Output** | Notifications , audit logs |
| **Business rules** | Notifications must be logged for compliance |
| **Miscellaneous Info’s** | Future integration with external messaging system |

|  |  |
| --- | --- |
| **Use Case Spec** | **Risk Management Alerts** |
| **Description** | System automatically flags risk based on project data |
| **Actors** | Project Managers, System |
| **Pre-conditions** | Project must have active tasks with deadlines |
| **Post Conditions** | Risks are logged and displayed on the dashboard |
| **Basic Flows** | 1. System monitor project progress 2. Identifies overdue tasks or potential delays 3. Flags risk and notifies the PM |
| **Alternate Flows** | User manually marks a task as High risk |
| **Exceptional Flows** | If risk detection fails, an alert is sent to admin |
| **Assumptions** | System has predefined risk criteria |
| **Constraints** | Risks assessment is based on available data |
| **Dependencies** | Requires tasks tracking and milestone monitoring |
| **Input** | Task completion rates, deadlines |
| **Output** | Risk alerts, notifications |
| **Business rules** | Risks are categorized based on severity |
| **Miscellaneous Info’s** | Users can customize risk thresholds |

|  |  |
| --- | --- |
| **Use Case Spec** | **Data Synchronization** |
| **Description** | Ensures project data is synced across multiple platforms |
| **Actors** | Admin, System |
| **Pre-conditions** | System must be connected to external databases |
| **Post Conditions** | Data is updated in real time across all platforms |
| **Basic Flows** | 1. System checks for data inconsistencies 2. Synchronizes project data 3. Confirms data integrity and updates logs |
| **Alternate Flows** | Manual synchronization can be triggered by admin |
| **Exceptional Flows** | If sync fails, an error is logged |
| **Assumptions** | All platforms use the same data format |
| **Constraints** | Synchronization depends on internet connectivity |
| **Dependencies** | Requires project tracking and database integration |
| **Input** | Project updates, external tool data |
| **Output** | Updated system wide data |
| **Business rules** | Data sync occurs every 5 minutes |
| **Miscellaneous Info’s** | Logs are maintained for audit purposes |

|  |  |
| --- | --- |
| **Use Case Spec** | **Historical Data Access** |
| **Description** | Allows users to access past project records |
| **Actors** | Project Managers, Upper management |
| **Pre-conditions** | Project must have completed milestones |
| **Post Conditions** | Historical records are retrieved successfully |
| **Basic Flows** | 1. User selects ”View historical data” 2. Filters records by data range 3. System retrieves and display data |
| **Alternate Flows** | If no data is found, a notification is displayed |
| **Exceptional Flows** | If database retrieval fails, an error is logged |
| **Assumptions** | Historical data is regularly backed up |
| **Constraints** | Data older than five years may be archived |
| **Dependencies** | Requires database access |
| **Input** | Project ID, data range |
| **Output** | Historical project records |
| **Business rules** | Only authorized users can access historical data |
| **Miscellaneous Info’s** | Reports can be generated from historical data |

|  |  |
| --- | --- |
| **Use Case Spec** | **Role based Access** |
| **Description** | Manages user permissions based on roles |
| **Actors** | Admin |
| **Pre-conditions** | User must have assigned roles |
| **Post Conditions** | Access rights are enforced |
| **Basic Flows** | 1. Admin assigns roles 2. Users login and system grants appropriate access |
| **Alternate Flows** | User request additional access |
| **Exceptional Flows** | Unauthorized access attempts are blocked |
| **Assumptions** | Roles are predefined |
| **Constraints** | Users cannot modify their own roles |
| **Dependencies** | Requires user authentication module |
| **Input** | User ID, role assignment |
| **Output** | Access granted or restricted |
| **Business rules** | Admin approval required for role changes |
| **Miscellaneous Info’s** | System logs access changes if required |

|  |  |
| --- | --- |
| **Use Case Spec** | **Data Encryption** |
| **Description** | Encrypt sensitive data in transit and rest |
| **Actors** | System, admin |
| **Pre-conditions** | Users must be authenticated |
| **Post Conditions** | System stored data securely |
| **Basic Flows** | 1. User inputs sensitive data 2. System encrypts data before storage 3. Decryption occurs upon authorized retrieval |
| **Alternate Flows** | Encryption settings are modified by admin |
| **Exceptional Flows** | Encryption fails system alerts security team |
| **Assumptions** | Encryption algorithms meet compliance standard |
| **Constraints** | Processing power limits performance |
| **Dependencies** | Requires secures key management |
| **Input** | User credentials, financial data |
| **Output** | Encrypted data logs |
| **Business rules** | Only authorized users can decrypt data |
| **Miscellaneous Info’s** | Encryption keys must be rotated periodically |

|  |  |
| --- | --- |
| **Use Case Spec** | **System Performance Monitoring** |
| **Description** | Monitor system performance to ensure efficiency |
| **Actors** | System, Admin |
| **Pre-conditions** | System must be actively running |
| **Post Conditions** | Performance reports are generated |
| **Basic Flows** | 1. System logs resource usage 2. Alerts are triggered if performance drops 3. Admin review logs |
| **Alternate Flows** | Admin manually checks system performance |
| **Exceptional Flows** | Monitoring tools fail, system logs error |
| **Assumptions** | Real time monitoring tools are active |
| **Constraints** | System monitoring consumes resources |
| **Dependencies** | Requires database and server logs |
| **Input** | CPU usage, memory consumption |
| **Output** | Performance reports and alerts |
| **Business rules** | Logs must be retained for compliance |
| **Miscellaneous Info’s** | Future AI driven optimizations planned |

|  |  |
| --- | --- |
| **Use Case Spec** | **Feedback Submission** |
| **Description** | Collects and manage stakeholder feedback |
| **Actors** | Project manager, Team member, Stakeholder |
| **Pre-conditions** | User must have access to feedback form |
| **Post Conditions** | Feedback is recorded and analyst |
| **Basic Flows** | 1. User access feedback form 2. Inputs comments and submits 3. System stores feedback for review |
| **Alternate Flows** | Feedback is submitted via email integration |
| **Exceptional Flows** | Feedback submission fails; system logs error |
| **Assumptions** | Users are willing to provide feedback |
| **Constraints** | Feedback is limited to text input |
| **Dependencies** | Requires database for storage |
| **Input** | User comments, ratings |
| **Output** | Feedback reports |
| **Business rules** | Anonymous feedback must be allowed |
| **Miscellaneous Info’s** | AI sentiment analysis may be integrated |

|  |  |
| --- | --- |
| **Use Case Spec** | **Messaging & Collaboration** |
| **Description** | Enable real time messaging between team members |
| **Actors** | Project Manager, Team member, stakeholders |
| **Pre-conditions** | User must be logged in |
| **Post Conditions** | Messages are delivered and stored |
| **Basic Flows** | 1. User opens chat interface and Send message 2. System delivers and logs message |
| **Alternate Flows** | User send file attachments via chat |
| **Exceptional Flows** | Message fails due to network issues, system retrieves |
| **Assumptions** | Users require real time communication |
| **Constraints** | Chat history is stored for limited time |
| **Dependencies** | Requires database for message logs |
| **Input** | Text messages, file attachments |
| **Output** | Message logs, notifications |
| **Business rules** | Confidential data should not be shared over chat |
| **Miscellaneous Info’s** | Future integration with external chat tools planned |

## Activity Diagrams

## Project Creation

## 

## Milestone & Task Tracking

## 

## Manual Progress Updates

## 

## Visual Dashboard

## 

## Generate Reports

## 

## Automated Notifications

## 

## VII Task Assignment and Dependencies

## 

## VIII Risk management alerts

## 

## IX Data Synchronization

## 

## X Historical Data Access

## 

## XI Role Based Access

## 

## XII Data Encryption

## 

## XIII System Performance Monitoring

## 

## XIV Feedback Submission

## 

## XV Message Collaboration

## 

## Document 7- Screens and pages

## Login Page

## 

## Home Page

## 

## Project Creation Page

## 

## Milestone Tracking

## 

## Manual Progress Update

## 

## Visual Dashboard

## 

## Generate Reports

## 

## Task Assignment and Dependencies

## 

## Automated Notifications and alerts

## 

## Risk Management

## 

## Data Synchronization

## 

## Historical Data Access

## 

## Role based Access

## 

## System Monitoring

## 

## Message Collaboration

## Feedback Submission

## 

## Document 8- Tools-Visio and Axure

1. **MS AZURE :**

It provides a suite of services and tools for Project management, development and deployment. Some of context of design tools like azure boards, devops, and clouds. Some of the features are, it creates dashboards to track project status and requirement. PM tool for tracking tasks, user stories and progress. In azure boards, which is an end to end development lifecycle management including testing. Works with tools like visual studio, and store and manage files, wireframes, and prototypes in cloud storage.

**Advantages** - It allows team collaboration in real time and communication. It supports all kinds of projects from any level of applications. Also, it enable remote access and seamless collaboration across teams

**Disadvantages –** It requires time to learn and set up for teams and costs huge. It may be too complex for small teams and some features pipelines and advanced workflows need expertise.

1. **Visio:**

It is a diagramming and vector graphics application used to create flowcharts, network diagram, floor plans and process maps. It is part of Microsoft office suite which provides a wide range of built in templates, shapes and stencils to simplify the creation of complex diagrams. It support s data linking, allowing users to connect diagrams with real time data sources for dynamic updates. It also offers collaboration features, enabling multiple users to work on the same diagram simultaneously. It is used in business, engineering and IT fields for visualizing and communicating information effectively

**Advantages –** Data connectivity features allow for real-time updates, enhancing efficiency in business and technical workflows. Its collaboration capabilities help teams work together seamlessly on projects.

**Disadvantages – It** is relatively expensive compared to other diagramming tools, making it less accessible for individuals and small businesses.

The learning curve can be steep for beginners, and compatibility issues may arise when sharing files with users who do not have Visio installed.

## Document 9- BA experience

#### My experience as BA in following phases:

1. **Requirement gathering:**

* Requirement gathering involves engaging with stakeholders to identify business needs, expectations, and objectives. I conducts various interviews, workshops, and brainstorming sessions to extract necessary details.
* Understanding business problems, constraints, and priorities is key to ensuring the project’s success. Various documents such as Business Requirement Document (BRD), stakeholder analysis, and meeting notes are created.
* Techniques like SWOT analysis, MoSCoW prioritization, and user interviews help refine requirements.
* While working with clients, it is crucial to ask the right questions and ensure alignment between their vision and business goals. Sometimes, clients have struggle to articulate their needs, requiring the BA to guide them in defining their objectives.
* Ensuring effective communication and capturing unspoken requirements plays a significant role in preventing scope creep.

#### Requirement Analysis:

* This phase involves critically evaluating and refining gathered requirements to ensure they are clear, complete, and feasible. We are able to identify gaps, conflicts, and dependencies between different requirements.
* Detailed analysis is conducted using techniques such as GAP analysis, data flow diagrams (DFD), and UML diagrams like Use case and activity diagram.
* Documents such as the Functional Requirement Document (FRD), process flow diagrams, and use cases are created to ensure developers and stakeholders have a shared understanding.
* While working with clients, challenges often arise when their requirements are ambiguous or conflicting. It is essential to collaborate with them to prioritize features based on business impact and feasibility.
* Regular review sessions help ensure alignment, and managing change requests effectively is necessary to keep the project on track.

#### Design:

* During the design phase, wecollaborates with UI/UX designers, architects, and developers to ensure that business requirements translate into a functional solution.
* This includes defining workflows, screen layouts, user experience guidelines, and system architecture. Wireframes, prototypes, and system design documents are created to provide a visual representation of the final product.
* Techniques such as storyboarding, mockups, and wire framing help validate designs with stakeholders before development begins.
* Client involvement is crucial at this stage, as they often request modifications based on usability and aesthetics. Frequent feedback loops are required to refine designs, and balancing business needs with technical feasibility is a common challenge.
* Clients may also have concerns about usability and scalability, requiring a BA to ensure that designs align with long-term business strategies.

#### Development:

* The development phase focuses on building the system according to the finalized requirements and design. We acts as a bridge between the development team and stakeholders, clarifying doubts and ensuring that the solution aligns with business objectives.
* Key documents such as user stories, API documentation, and data mapping sheets guide the development team.
* Clients may request changes during development, leading to scope adjustments and re-evaluations of priorities. It is essential to balance feasibility with business value while keeping project timelines in check.
* Sometimes, misunderstandings between business users and developers arise, requiring the BA to facilitate communication and resolve discrepancies quickly.

#### Testing:

* In the testing phase, we works closely with the QA team to validate that the developed solution meets the defined requirements. They review test cases, participate in defect triaging, and ensure that all business scenarios are covered.
* Techniques such as functional testing, regression testing, and user acceptance testing (UAT) are conducted. UAT plays a vital role in ensuring that the final product aligns with user expectations.
* We assists clients in conducting UAT by providing test scenarios, resolving queries, and documenting feedback. Clients often raise concerns about usability, performance, and unexpected behaviors, requiring quick resolutions before deployment. Managing last-minute changes while ensuring quality is a key challenge at this stage.

#### Deployment:

* The deployment phase involves releasing the solution to production while ensuring minimal disruption to business operations. We coordinates with IT, business users, and support teams to execute a smooth go-live process
* Release notes, training manuals, and support documentation are created to help end users adopt the new system. Techniques such as change management, go-live planning, and rollback strategies are used to minimize risks.
* Clients expect a seamless transition and often require hands-on support during this period. Addressing last-minute concerns, training business users, and ensuring a smooth knowledge transfer process are critical responsibilities.
* Unexpected issues may arise post-deployment, requiring quick fixes and proactive communication with stakeholders to maintain confidence in the solution.