XcelTrack

Software Requirement Specification

FYP Team

Abdul Basit (22F-3834)

Rehana Hassan (22F-3819)

Maleeha Batool (22F-3372)

Supervised By:

Mr. Rizwan Ul Haq

Department of Computer Science

National University of Computer and Emerging Science

Chiniot, Pakistan

2025

Table of Contents

List of Tables:

List of Figures:

1. Introduction

XcelTrack is an innovative project designed to revolutionize how professionals collaborate on Microsoft Excel spreadsheets. This document outlines the complete software requirements for developing a system that brings modern version control principles, similar to Git, directly to Excel workflows. It aims to solve critical problems of tracking changes, managing collaboration, and ensuring data integrity in data analysis and financial modeling.

2. Vision Document

2.1 Problem Statement:

Table 1: Problem Statement

Category	Description
Problem	• Reliance on manual versioning (e.g., "File_final_v2.xlsx") leading to confusion and lost work. • Inability to track who changed what, when, and why in a collaborative environment. • No native support for merging concurrent changes from multiple users, causing conflicts and data loss. • Difficulty in auditing spreadsheet history for compliance and debugging purposes.
Affects	• Impacts financial analysts, data scientis • ts, accountants, and business professionals who rely on Excel for critical decision-making. • Hinders team productivity and erodes trust in data accuracy within organizations. • Makes regulatory compliance and internal auditing processes tedious and error-prone.
Impact	• Results in significant time wasted reconciling different spreadsheet versions. • Leads to flawed business decisions based on incorrect or outdated data. • Creates financial and reputational risk due to undetected errors and lack of a clear audit trail.
Solution	A web-based platform that provides: • **Git-like Version Control for Excel:** Track every change at the individual cell level with commit messages. • **Intelligent Collaboration:** Detect conflicts and provide tools for seamless merging. • **AI-Powered Insights:** Explain formulas and detect errors automatically. • **Hybrid Workflow Support:** Synchronize changes between online and desktop editing.

2.2 Business Opportunities:

XcelTrack addresses a significant gap in a market dominated by Microsoft Excel, which has over a billion users worldwide. The system presents substantial opportunities for commercialization, including a subscription-based Software-as-a-Service (SaaS) model for businesses and enterprises. Potential also exists for partnerships with educational institutions for teaching data management and with corporations seeking to enhance their data governance and compliance frameworks.

2.3 Objectives:

Following are the objectives of XcelTrack:

- To develop a secure, web-based platform for uploading, managing, and version-controlling Excel files.
- To implement a robust version control system that tracks changes to values, formulas, and formatting at the individual cell level.
- To generate human-readable "semantic diffs" that clearly explain what changed in each commit (e.g., "Formula in B2 changed from SUM to AVERAGE").
- To integrate AI services for automatic formula explanation, error detection, and data anomaly identification.
- To enable real-time and offline collaboration with intelligent conflict detection and resolution tools.
- To provide a comprehensive admin dashboard for user management, system monitoring, and generating compliance reports.

2.4 Scope:

The scope of XcelTrack encompasses the development of a full-stack web application. The core system will include user authentication, file management, the version control engine, the AI analysis module, and the admin dashboard. The initial release will focus on the web platform, with parsing and synchronization support for desktop Excel files (.xlsx). Features like a dedicated mobile app or real-time co-editing (like Google Sheets) are considered out of scope for the initial version.

2.5 Constraints:

The constraints are given below:

- The system's accuracy in parsing and diffing is dependent on the capabilities of the chosen JavaScript Excel library (e.g., SheetJS, ExcelJS).
- The performance of the AI features (explainability, error detection) is constrained by the chosen AI model/API and its associated costs and latency.
- The system will be designed as a web application and will not be a native Microsoft Excel plugin, requiring users to use the web interface or a sync-and-upload workflow.

2.6 Stakeholder and User Descriptions:

2.6.1 Market Demographics:

The target market includes business professionals, financial analysts, data scientists, researchers, accountants, and students across various industries such as finance, consulting, healthcare, and academia. Essentially, any individual or team that uses Excel for complex, collaborative, and auditable work is a potential user.

2.6.2 User Environment:

Users will primarily access XcelTrack through modern web browsers on desktops and laptops. The platform will be responsive but optimized for larger screens to accommodate complex spreadsheet UIs. The system will also support a hybrid environment where users can download files, edit them in the native Microsoft Excel desktop application, and then re-upload them to sync changes.

2.6.3 Stakeholder Profiles:

2.6.3.1 Supervisor:

Table 2: Supervisor

Representatives	Mr. Rizwan Ul Haq
Description	Provides academic and technical guidance for the project's development.
Type	Technical Stakeholder.
Responsibility	1. Provides direction to the development team. 2. Ensures the project is completed within the allocated timeframe. 3. Reviews the technical soundness of the proposed solution. 4. Tracks project progress and ensures adherence to the design documentation.
Success Criteria	The successful development and demonstration of a functional prototype that meets the core objectives.
Involvement	High 1. Requirement reviewer 2. Senior managers senior managers 3. Advisor throughout the project lifecycle

2.6.3.2 Development Team:

Table 3: Developer Team

Representatives	Abdul Basit, Rehana Hassan, Maleeha Battol
Description	The team responsible for the research, design, development, and testing of the XcelTrack system.
Type	Technical Stakeholder.
Responsibility	1. Conduct requirement analysis and system design. 2. Implement the frontend, backend, and database components. 3. Integrate third-party libraries and AI services.

	4. Test the system and document the development process.
Success Criteria	The successful deployment of a fully functional XcelTrack system as per the SRS.
Involvement	High - fully involved in all development phases.

2.6.3.3 End Users:

Table 4: End Users

Representatives	Financial Analysts, Data Scientists, Business Professionals
Description	The primary users who will use XcelTrack for their daily spreadsheet work..
Type	External Stakeholders.
Responsibility	Use the system to version control, collaborate on, and audit their Excel files. Provide feedback for improvements.
Involvement	Medium - they are the consumers of the final product.

2.6.3.4 Administrator:

Table 5: Administrator

Representatives	IT Managers, Team Leads
Description	Responsible for managing the XcelTrack instance within an organization.
Type	Internal Stakeholder
Responsibility	1. Manage user accounts and permissions. 2. Monitor system usage and performance. 3. Access and generate audit and compliance reports.
Involvement	Low to Medium - they interact with the admin dashboard.

2.6.4 Stakeholder Summary:

Table 6: Stakeholder Summary

Name	Description	Responsibility
Development Team	The team building the XcelTrack system.	Research, design, develop, test, and deploy the entire system
Supervisor Team	Provides academic and technical guidance.	Guide the team, review progress, and ensure project quality.
End User	The professional using the system for spreadsheet work	Use the system's core features for version control and collaboration.
Admins	Manages the platform for an organization.	Manage users, monitor system health, and generate reports.

3. System Requirements Specification:

3.1 System Features:

XcelTrack offers a range of critical features, including:

- **User Authentication and Authorization:** Secure login and role-based access control for users and admins.
- **Workbook and Cell-Level Version Control:** A Git-like system that tracks every change to a spreadsheet, allowing users to commit changes with messages and revert to any previous version of a cell or the entire workbook.
- **Worksheet Management**: Comprehensive control over workbook structure, allowing users to create, rename, reorder, and delete worksheets while maintaining version history and collaboration integrity.
- **Semantic Diffs and Audit Trail:** Generates human-readable summaries of changes (diffs) and maintains a complete, tamper-evident history of all user actions.

- **AI-Powered Intelligence:** Automatically explains complex formulas in plain language, detects potential errors and anomalies, and provides smart recommendations.
- **Collaboration and Merge Tools:** Allows multiple users to work on the same file and provides tools to detect and resolve edit conflicts.
- **Hybrid Editing Support:** Supports both online editing in the web app and offline editing in Microsoft Excel Desktop, with seamless synchronization upon re-upload.
- **Admin and Compliance Dashboard:** Provides a central interface for user management, system monitoring, and generating detailed audit reports.

3.2 Functional Requirements:

The system must fulfil the following functional requirements:

3.2.1 User Authentication and Authorization:

FR1.1 The system shall allow users to create an account using email and password.

FR1.2 The system shall allow users to log in securely using authentication mechanisms.

FR1.3 The system shall authorize users based on their roles (e.g., Admin, User).

FR1.4 The system shall restrict access to project data based on assigned roles and permissions.

FR1.5 The system shall provide password reset and account recovery features.

3.2.2 Excel File Management:

FR2.1 The system shall allow users to upload Excel files (.xlsx, .xls).

FR2.2 The system shall use SheetJS/ExcelJS library to parse uploaded Excel files at cell level.

FR2.3 The system shall store parsed data and metadata in backend database.

FR2.4 The system shall allow users to download Excel files with all recent changes merged.

FR2.5 The system shall maintain file metadata (owner, creation date, version, size).

3.2.3 Cell-Level Version Control:

FR3.1 The system shall record every change made to individual cells as separate versions.

FR3.2 Each version shall be tagged with timestamp, user ID, and commit message.

FR3.3 The system shall allow users to view complete version history of workbook or single cell.

FR3.4 The system shall allow users to revert to previous version of entire workbook or individual cell.

FR3.5 The system shall support commit-based version history similar to Git.

3.2.4 Semantic Diffs and Audit Trail:

FR4.1 The system shall generate human-readable summaries of changes instead of raw code differences.

FR4.2 The system shall display semantic diffs (e.g., "Formula in B2 changed from SUM to AVERAGE").

FR4.3 The system shall maintain tamper-proof audit log of all edits for compliance.

FR4.4 The system shall allow users to filter and search audit trail by user, date, cell reference, or commit message.

3.2.5 AI-Powered Assistance:

FR5.1 The system shall automatically detect common Excel errors.

FR5.2 The system shall use AI to analyze data patterns and highlight potential outliers.

FR5.3 The system shall use AI to suggest conflict resolution options during merges.

FR5.4 The system may suggest alternative, more efficient functions for simple formulas.

3.2.6 Real-time Collaboration:

FR6.1 The system shall allow multiple users to edit same spreadsheet simultaneously through web interface.

FR6.2 The system shall detect when two or more users modify same cell at same time.

FR6.3 The system shall flag conflicting edits and display them to users for resolution.

FR6.4 The system shall provide merge tools allowing users to accept or reject changes intelligently.

FR6.5 The system shall support real-time synchronization of edits across all active sessions.

3.2.7 Hybrid Editing Support:

FR7.1 The system shall allow users to download Excel files for offline editing.

FR7.2 When offline file is re-uploaded, system shall automatically parse all changes.

FR7.3 The system shall integrate offline changes into main version history.

FR7.4 The system shall detect conflicts between offline and online edits and notify users.

FR7.5 The web client shall maintain offline edits in browser's local storage until re-upload.

3.2.8 Notifications System:

FR8.1 The system shall prompt users to add commit messages for major edits.

FR8.2 The system shall send notifications (in-app or email) for important events.

FR8.3 The system shall provide notification panel for quick updates on file changes.

3.2.9 Admin and Compliance Dashboard:

FR9.1 The system shall provide Admin dashboard for monitoring platform usage.

FR9.2 The Admin shall be able to view logs, user activity, and system status.

FR9.3 The Admin shall manage user accounts, permissions, and access levels.

FR9.4 The system shall allow generation of compliance reports in downloadable format.

FR9.5 The system shall enable audit-ready logs for regulated industries.

3.2.10 System Integration & APIs:

FR10.1 The system shall use Univer API to provide online spreadsheet editing interface.

FR10.2 The system shall provide internal APIs for frontend-backend communication.

FR10.3 The system shall support integration with third-party services if required.

3.2.11 Worksheet Management Requirements:

FR11.1 The system shall allow users to create new worksheets within an existing workbook.

FR11.2 The system shall allow users to rename existing worksheets.

FR11.3 The system shall allow users to reorder worksheets within a workbook by changing their position index.

FR11.4 The system shall allow users to delete worksheets from a workbook, with appropriate confirmation prompts.

FR11.5 The system shall track worksheet-level changes (creation, renaming, deletion, reordering) in the version history.

FR11.6 The system shall maintain worksheet metadata including name, position index, creation date, and last modification date.

FR11.7 The system shall validate worksheet names to prevent duplicates within the same workbook.

FR11.8 The system shall prevent deletion of the last remaining worksheet in a workbook.

FR11.9 The system shall support worksheet operations in both online and hybrid editing modes.

FR11.10 The system shall synchronize worksheet changes across all active collaboration sessions in real-time.

3.3 Non-Functional Requirements:

NFR1. Performance:

- The system shall handle Excel files up to 10 MB with less than 2 seconds response time for upload and version diff generation.
- The system shall support simultaneous editing by at least 10 users without noticeable lag.
- The system shall maintain efficient query performance for version history and diffs.

NFR2. Reliability & Availability:

- The system shall have uptime of at least 99% during normal operation.

- The system shall implement error handling and recovery to avoid data loss during merges or conflicts.
- In case of failure, system state should be recoverable from database backups.

NFR3. Security:

- All communications between client and server shall be encrypted using HTTPS/TLS.
- User passwords shall be hashed and salted before storage.
- The system shall implement role-based access control to prevent unauthorized access.
- The system shall log and monitor suspicious activities for auditing.

NFR4. Compatibility:

- The system shall work on modern browsers (Chrome, Firefox, Edge).
- The system shall integrate smoothly with React.js frontend and Node.js backend.

NFR5. Usability:

- The user interface shall be intuitive, with clean dashboard and easy navigation.
- The system shall provide clear error messages and tooltips for guidance.
- The interface shall support responsive design for various screen sizes.

NFR7. Maintainability:

- The system shall use Git for version control of source code.
- The system shall support Dockerized environments for easy deployment.

4. Design Artifacts:

4.1. Use Case Diagram

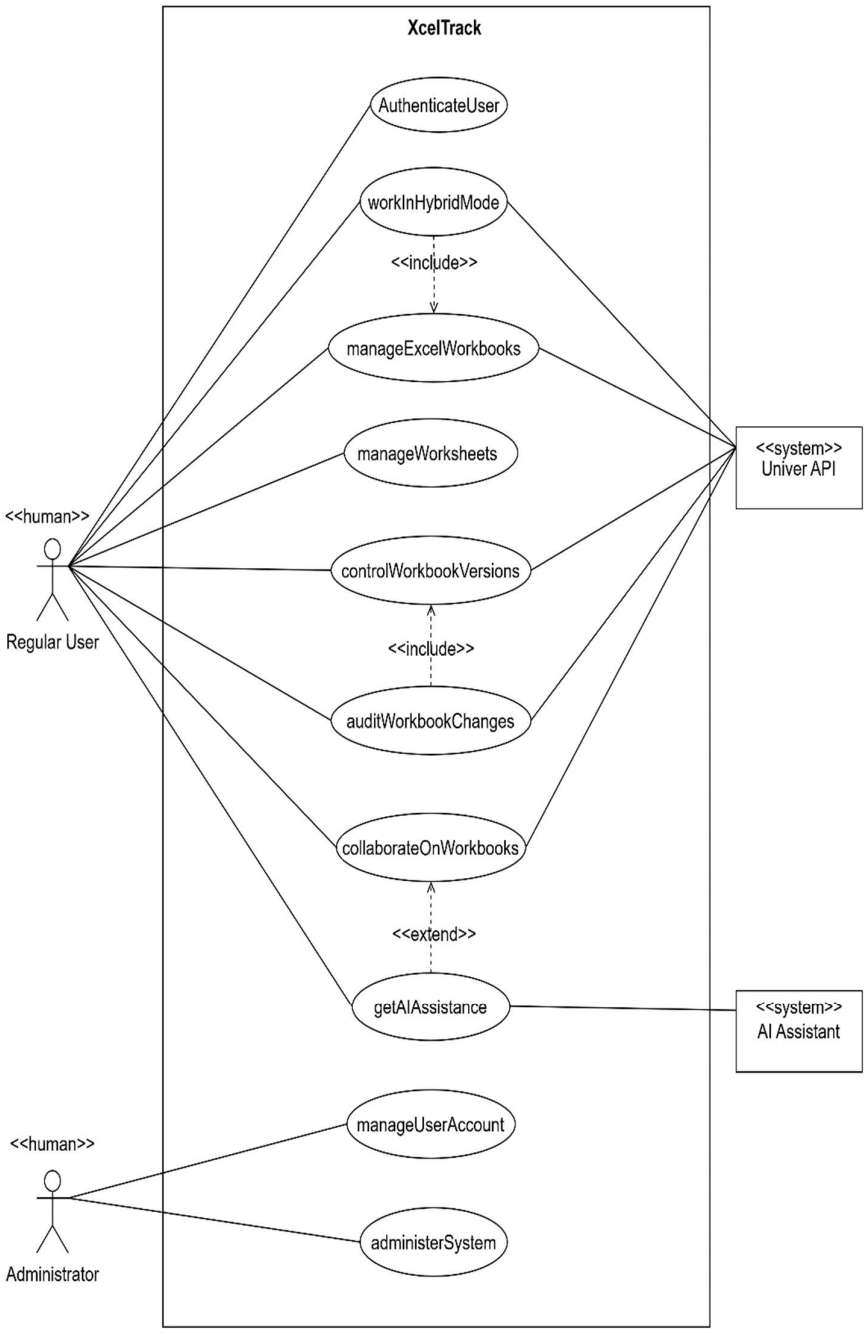

Figure 1: Use Case Diagram

4.2. Expanded Use Cases

4.2.1. User Authentication

Table 7: User Authentication

Section	Content
Designation	UC-01
Name	User Authentication
Actors	User (Admin, Editor, Viewer)
Priority	High
Criticality	Essential for secure system access
Source	Functional Requirements(3.2.1)
Description	Enables users to register, log in, or recover accounts securely to access the XcelTrack platform.
Trigger Event	The user attempts to access or create an account.
Action	The user provides credentials to log in, registration details to sign up, or initiates password recovery.
Pre-conditions	1. The user has a valid email address. 2. For registration, the email is not already registered. 3. For login, the account must exist.
Post-conditions	1. A new user account is created. 2. The user is authenticated and redirected to their role-based dashboard. 3. The user's password is successfully updated.
Results	Secure, role-based access to the XcelTrack application.

Main Scenario	2.1 The user registers or logs in by providing valid credentials. 2.2 The system validates the input, authenticates the user, and redirects them to their role-based dashboard.
Alternative Scenario	1A. The user selects "Forgot Password" to recover access. 1A.1 The system sends a password reset link to the registered email. 1A.2 The user resets the password using the link. 1A.3 The system confirms the password update and redirects to the login page. 1B. The user has previously enabled "Remember Me". 1B.1 The system automatically authenticates the user. 1B.2 The system redirects the user to their role-based dashboard.
Exception Scenario	1a. If the registration email already exists or a password reset is requested for an unregistered email, 1a.1 The system notifies the user that the email is already registered or invalid for reset. 2a. If the login credentials are invalid, 2a.1 The system displays a generic error message without specifying which field is incorrect.

4.2.2. Manage Excel Workbooks

Table 8: Manage Excel Workbooks

Section	Content
Designation	UC-02
Name	Manage Excel Workbooks
Actors	User (Admin, Editor, Viewer)

Priority	High
Source	Functional Requirements(3.2.2)
Description	Allows users to upload, parse, store, download, and synchronize Excel workbooks with cell-level history, metadata, and merged changes.
Trigger Event	The user uploads, downloads, or synchronizes an Excel workbook.
Action	The system parses uploaded files at cell level, stores parsed content and metadata, and produces downloadable workbooks with merged changes.
Pre-conditions	1. The user is authenticated and authorized to access the workspace. 2. The file is in a supported Excel format (.xlsx, .xls).
Post-conditions	1. Parsed workbook data and metadata are stored in the database. 2. The user can download a workbook with recent changes merged. 3. Cell-level history and metadata (owner, creation date, version, size) are recorded.
Results	Persistent, versioned storage of workbook content at cell level and correct upload/download/merge behavior.
Main Scenario	1. The user uploads a supported Excel file. 2. The system parses the workbook at the cell level (values, formulas, formatting) using the configured library and stores parsed data and metadata in the backend. 3. The user requests a download; the system generates and returns an Excel file with recent changes merged.
Alternative Scenario	1A. The user uploads a workbook edited offline. 1A.1 The system parses the offline changes. 1A.2 The system merges them into the version history.

	1A.3 The system prompts the user to resolve conflicts if detected. 3A. The user requests to download a previous version of the workbook. 3A.1 The system retrieves the selected version from the version history. 3A.2 The system generates the corresponding Excel file.
Exception Scenario	1a. If the uploaded file is unsupported or corrupted, the system rejects it and notifies the user. 2a. If a parsing or storage error occurs, the system reports the failure and suggests the user retry or contact support.

4.2.3. Control Workbook Versions

Table 9: Control Workbook Versions

Section	**Content**
Designation	UC-03
Name	Control Workbook Versions
Actors	User (Admin, Editor, Viewer)
Priority	High
Source	Functional Requirements (3.2.3)
Description	Enables users to track, review, and manage cell-level version history of Excel workbooks, including commits, history viewing, and rollback operations.
Trigger Event	The user modifies a workbook, commits changes, or requests to view or restore a previous version.
Action	The system captures each cell change as a version entry with timestamp, user ID, and commit message, maintains version history, and supports restoration or comparison of versions.

Pre-conditions	1. The user is authenticated and has edit or view access to the workbook. 2. The workbook is already uploaded and stored in the system.
Post-conditions	1. All cell modifications are recorded as new version entries. 2. Version history is updated with timestamp, user ID, and commit message. 3. The user can successfully view or revert to a previous version.
Results	The system maintains a detailed, cell-level version history and enables rollback or comparison operations .
Main Scenario	1. The user edits one or more cells in the workbook and saves or commits changes. 2. The system records modified cells as a new version with timestamp, user ID, and commit message. 3. The user views the version history for a workbook or specific cell. 4. The system displays the chronological list of commits and changes.
Alternative Scenario	2A. The user selects a previous version and chooses to revert. 2A.1 The system restores the selected version. 2A.2 The system updates the current workbook state. 3A. The user compares two versions of the workbook. 3A.1 The system highlights the differences between the selected versions at the cell level.
Exception Scenario	2a. If a version record is missing or corrupted, the system notifies the user and prevents rollback. 2b. If the user lacks permission to modify versions, the system restricts the action and displays an access warning.

4.2.4. Audit Workbook Changes

Table 10: Audit Workbook Changes

Section	Content
Designation	UC-04
Name	Audit Workbook Changes
Actors	User (Admin, Editor, Viewer)
Priority	High
Source	Functional Requirements(3.2.4)
Description	Enables users to review, search, and verify all modifications made to Excel workbooks through semantic diffs and a secure audit trail.
Trigger Event	The user requests to view recent workbook changes or access the audit trail.
Action	The system generates readable summaries of edits, stores audit entries securely, and allows filtering and search by user, date, cell, or commit message.
Pre-conditions	1. The user is authenticated. 2. Version history exists for the workbook.
Post-conditions	1. The system produces human-readable summaries of cell-level changes. 2. All actions are logged in an immutable audit trail. 3. The user can locate specific changes using filters or search.
Results	Transparent, tamper-proof record of workbook activity that supports accountability and compliance.
Main Scenario	1. The user opens the workbook's change history or audit panel. 2. The system retrieves and displays semantic diffs for each change.

	3. The user reviews the summarized changes in chronological order.
Alternative Scenario	3A. The user applies filters to the audit trail (e.g., by user, date, or cell reference). 3A.1 The system displays only the matching audit entries. 3B. The user searches for a keyword or commit message. 3B.1 The system highlights and lists the corresponding results.
Exception Scenario	1a. If the audit log is unavailable or corrupted, the system notifies the user and restricts access until it is restored. 3a. If no changes match the search or filter criteria, the system displays a "No results found" message.

4.2.5. Get AI Assistance

Table 11: Get AI Assistance

Section	Content
Designation	UC-05
Name	Get AI Assistance
Actors	User (Admin, Editor, Viewer)
Priority	Medium-High
Source	Functional Requirements(3.2.5)
Description	Provides AI-driven insights, explanations, and correction suggestions for Excel workbooks to assist users in error detection, formula understanding, anomaly analysis, and conflict resolution.
Trigger Event	The user requests AI help or the system detects an issue that triggers automatic analysis.

Action	The system analyzes cell data and formulas using AI models, detects errors or anomalies, and presents contextual explanations or correction suggestions.
Pre-conditions	1. The workbook is uploaded and parsed. 2. The user is authenticated and authorized to access AI features.
Post-conditions	1. The system provides AI-generated insights or suggestions to the user. 2. Detected issues are logged for reference or further action. 3. The user may apply or dismiss the suggested corrections.
Results	Improved data accuracy, clearer formula understanding, and enhanced user productivity through AI-based guidance.
Main Scenario	1. The user opens a workbook and selects the AI Assistance option. 2. The system scans for common Excel errors (e.g., #REF!, #DIV/0!, circular references). 3. The system identifies complex formulas and generates natural-language explanations. 4. The user reviews the AI-provided insights and applies corrections or accepts suggestions.
Alternative Scenario	2A. The system automatically detects anomalies or inconsistent cell values during editing, alerts the user, and recommends review or correction. 4A. During merge or conflict handling, the system suggests a resolution 4A.1 (e.g., prioritize most recent change or a specific user's edit). 4B. The AI model suggests optimized formulas or functions 4B.1 (e.g., SUMIF instead of nested IFs) based on detected patterns.

Exception Scenario	1a. If AI services are unavailable, the system displays a message and proceeds with standard non-AI validation. 2a. If no issues or improvements are detected, the system informs the user that the workbook appears consistent.

4.2.6. Collaborate on Workbooks

Table 12: Collaborate on Workbooks

Section	Content
Designation	UC-06
Name	Collaborate on Workbooks
Actors	Multiple Users (Admin, Editor, Viewer)
Priority	High
Source	Functional Requirements(3.2.6)
Description	Enables multiple users to edit the same spreadsheet simultaneously through a web interface, supporting real-time synchronization, conflict detection, and intelligent merge resolution.
Trigger Event	Two or more users access and edit the same spreadsheet concurrently.
Action	The system synchronizes real-time edits, detects conflicting cell modifications, and provides merge tools for resolving differences.
Pre-conditions	1. The users are authenticated and have edit permissions for the same spreadsheet. 2. The spreadsheet is open in collaborative mode.
Post-conditions	1. All authorized edits are synchronized across user sessions in real time.

	2. Conflicting edits are detected and resolved. 3. The system updates the version history with merged changes.
Results	Seamless real-time collaboration with consistent data integrity across all active sessions.
Main Scenario	1. Multiple users open the same spreadsheet for editing. 2. The system establishes real-time communication and synchronizes all visible changes. 3. When a user edits a cell, the system updates all connected sessions instantly. 4. The system logs edits and maintains live consistency of shared data.
Alternative Scenario	3A. Two users edit the same cell simultaneously. 3A.1 The system detects the conflict and displays both versions to each user. 3A.2 Users review the differences and use merge options to accept or reject changes. 3B. The system automatically prioritizes the latest valid edit or user-selected preference 3B.1 when no manual action is taken.
Exception Scenario	1a. If unauthorized users attempt to edit the spreadsheet, 1a.1 the system denies access and displays a permission warning. 2a. If real-time synchronization fails due to network or server issues, 2a.1 the system temporarily stores local changes and re-syncs when the connection is restored.

4.2.7. Work in Hybrid Mode

Table 13: Work in Hybrid Mode

Section	Content
Designation	UC-07
Name	Work in Hybrid Mode
Actors	User (Admin, Editor, Viewer)
Priority	High
Source	Functional Requirements(3.2.7)
Description	Allows users to continue working on spreadsheets while offline and later synchronize their edits with the online version, ensuring consistent version control and conflict detection.
Trigger Event	The user downloads a spreadsheet for offline editing or reconnects to upload the updated version.
Action	The system enables offline storage and editing of workbooks, parses re-uploaded files for changes, and merges offline edits into the main version history upon synchronization.
Pre-conditions	1. The user is authenticated and authorized to access the spreadsheet. 2. The spreadsheet has been downloaded or cached for offline use.
Post-conditions	1. Offline edits are successfully synchronized with the online version. 2. Detected conflicts are highlighted and resolved by the user. 3. The system updates the version history and metadata accordingly.
Results	Users can edit spreadsheets without connectivity and later sync all changes safely with preserved version control.

Main Scenario	1. The user downloads the spreadsheet for offline work. 2. The user edits data while offline; changes are temporarily stored in local storage or Indexed DB. 3. When reconnected, the user uploads or re-syncs the spreadsheet. 4. The system parses all offline changes and merges them into the main version history.
Alternative Scenario	3A. The user reconnects after making offline edits. 3A.1 The system automatically detects the offline changes and prompts the user to confirm synchronization. 3B. The user chooses to preview differences before merging. 3B.1 The system displays a comparison of offline and online versions for review prior to synchronization.
Exception Scenario	3a. If the re-uploaded file is corrupted or incompatible, 3a.1 the system notifies the user and rejects synchronization. 4a. If conflicting edits exist, 4a.1 the system flags them and requests manual resolution before merging.

4.2.8. Administer System

Table 14: Administer System

Section	**Content**
Designation	UC-08
Name	Administer System
Actors	Admin
Priority	High
Source	Functional Requirements(3.2.9)

Description	Allows the Admin to oversee system operations, manage users, and monitor key events through commit prompts, notifications, and update panels for improved control and transparency.
Trigger Event	A system-level activity occurs, such as a major edit, conflict, merge, or user action requiring administrative visibility.
Action	The system prompts users for commit messages during significant edits, generates notifications for critical events, and updates the Admin's notification panel in real time.
Pre-conditions	1. The Admin is authenticated with full privileges. 2. The system is active and event monitoring is enabled.
Post-conditions	1. Commit messages for major edits are recorded. 2. Notifications are delivered for key actions such as merges or uploads. 3. The Admin dashboard displays recent updates and system status.
Results	The Admin maintains control and awareness of user activities, ensuring accountability and smooth system operation.
Main Scenario	1. A user performs a major edit in a spreadsheet. 2. The system prompts the user to enter a commit message. 3. The system records the commit and sends a notification to the Admin dashboard. 4. The Admin views all recent system activities and status updates.
Alternative Scenario	3A. The system sends additional alerts via email or in-app messages for critical operations such as conflicts or merges. 4A. The Admin filters notifications by category (e.g., uploads, edits, merges) to focus on specific activities.
Exception Scenario	2a. If a user bypasses the commit prompt, 2a.1 the system blocks submission until a message is provided.

	3a. If the notification or commit logging service fails, 3a.1 the system queues pending events for retry and informs the Admin of the issue.

4.2.9. Manage Worksheet

Table 15: Manage Worksheets

Section	Content
Designation	UC-09
Name	Worksheet Management
Actors	Editor, Collaborator
Priority	High
Source	Functional Requirements(3.2.11)
Description	Enables users to manage worksheets by creating, renaming, reordering, and deleting them within a workbook. The system validates names, maintains metadata, logs all changes, and synchronizes updates in real time across collaboration sessions.
Trigger Event	The user initiates a worksheet-level action (create, rename, reorder, delete).
Action	The system provides an interface for worksheet operations, validates user actions, prompts for confirmation where necessary, logs all changes to version history, and synchronizes updates across all active sessions in real-time.
Pre-conditions	1. The user is authenticated and has edit permissions for the target workbook. 2. The workbook is open in a compatible editing mode (Online or Hybrid). 3. The worksheet management module is active and functional.

Post-conditions	1. The requested worksheet operation is completed and reflected in the user's view. 2. Worksheet metadata (name, index, dates) is updated. 3. The change is recorded in the version history. 4. All collaborative sessions viewing the workbook are synchronized.
Results	The workbook's worksheet structure is successfully modified according to the user's intent, with changes persisted, tracked, and propagated to ensure consistency for all users.
Main Scenario	1. The user initiates a worksheet action such as create, rename, reorder, or delete. 2. The system validates the request, updates worksheet metadata, and performs the operation. 3. The system records the change in version history and synchronizes it across all active collaboration sessions.
Alternative Scenario	2A. If a worksheet name conflicts or is invalid, the system prompts the user to enter a corrected name. 3A. In hybrid mode, worksheet changes are temporarily stored and synchronized once connectivity is restored.
Exception Scenario	2a. If the user attempts to delete the last worksheet or enter an invalid name, the system blocks the action and displays an error. 3a. If the operation or synchronization fails, the system logs the issue and queues the change for retry.

4.1. Activity Diagram

4.1.1. Authenticate User

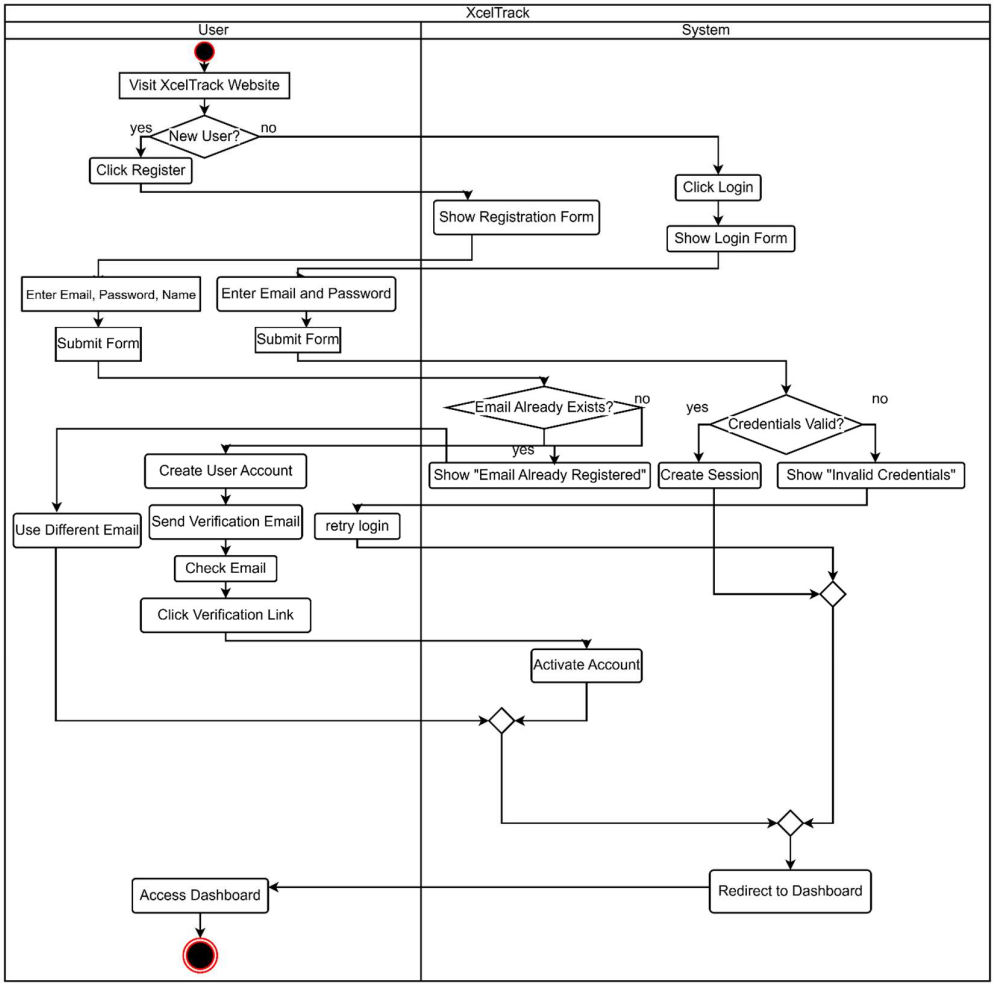

Figure 2: Authenticate User (Activity Diagram)

4.1.2. Manage excel Workbooks & Control Workbook Versions

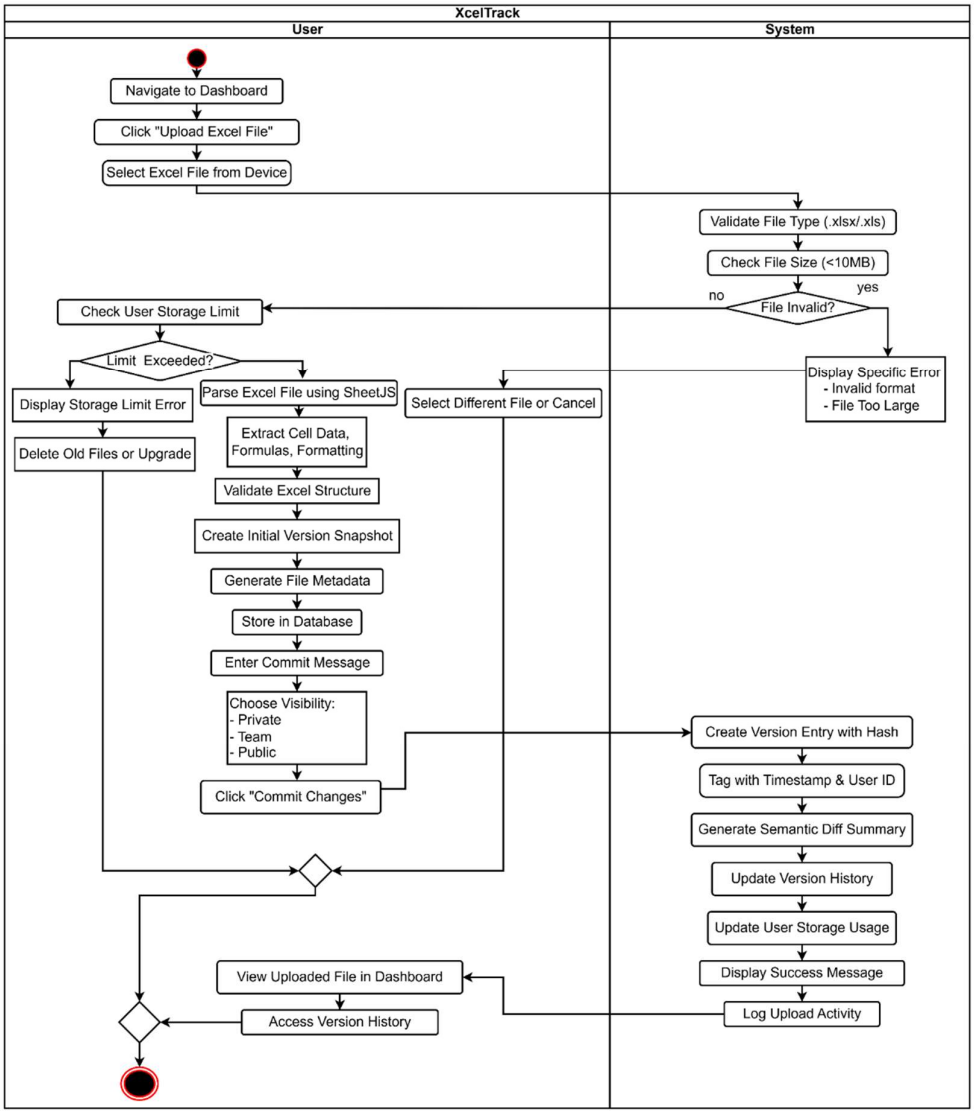

Figure 3: Manage excel Workbooks & Control Workbook Versions (Activity Diagram)

4.1.3. Audit Workbook Changes

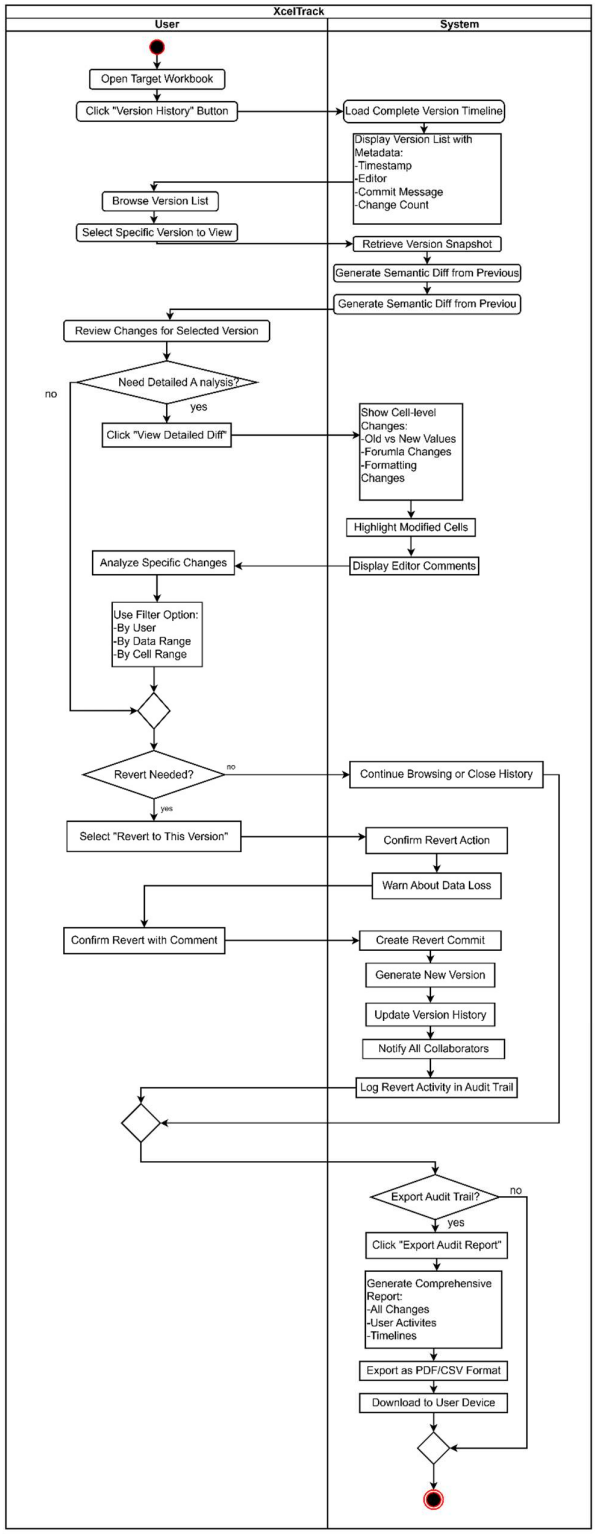

Figure 4: Audit Workbook Changes (Activity Diagram)

4.1.4. Get AI Assistance

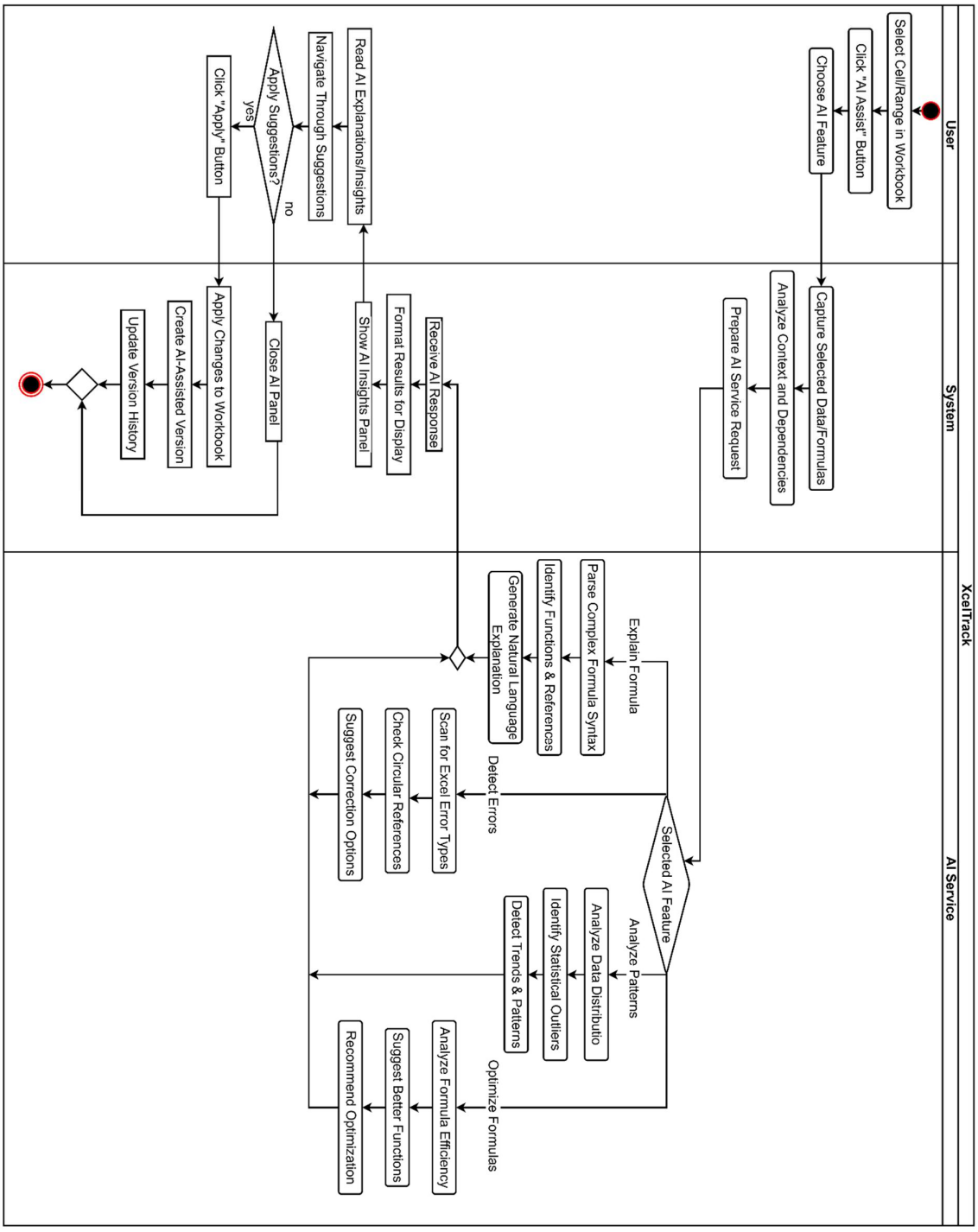

Figure 5: Get AI Assistance (Activity Diagram)

4.1.5. Collaborate on Workbooks

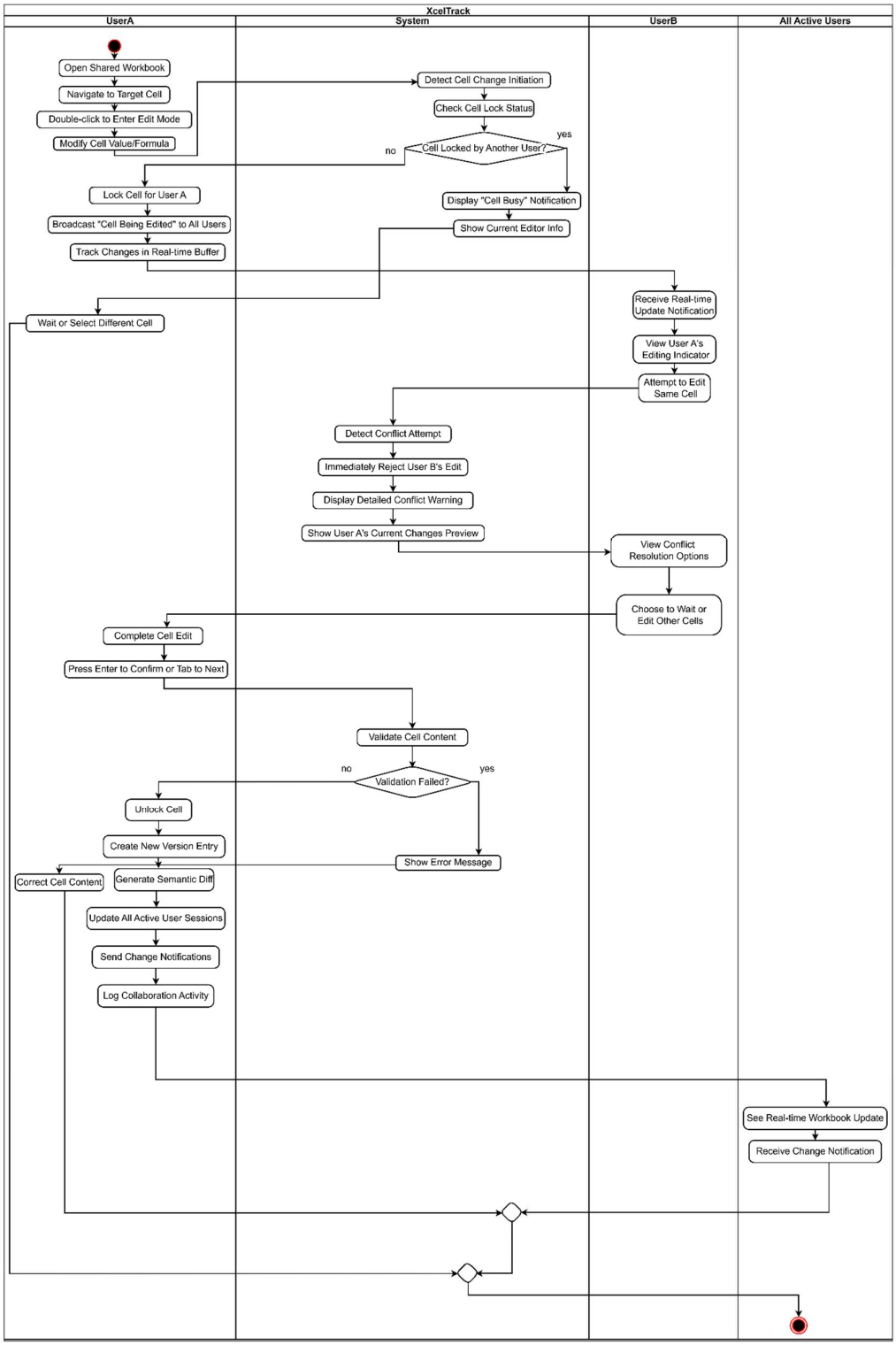

Figure 6: Collaborate on Workbooks (Activity Diagram)

4.1.6. Work in Hybrid Mode

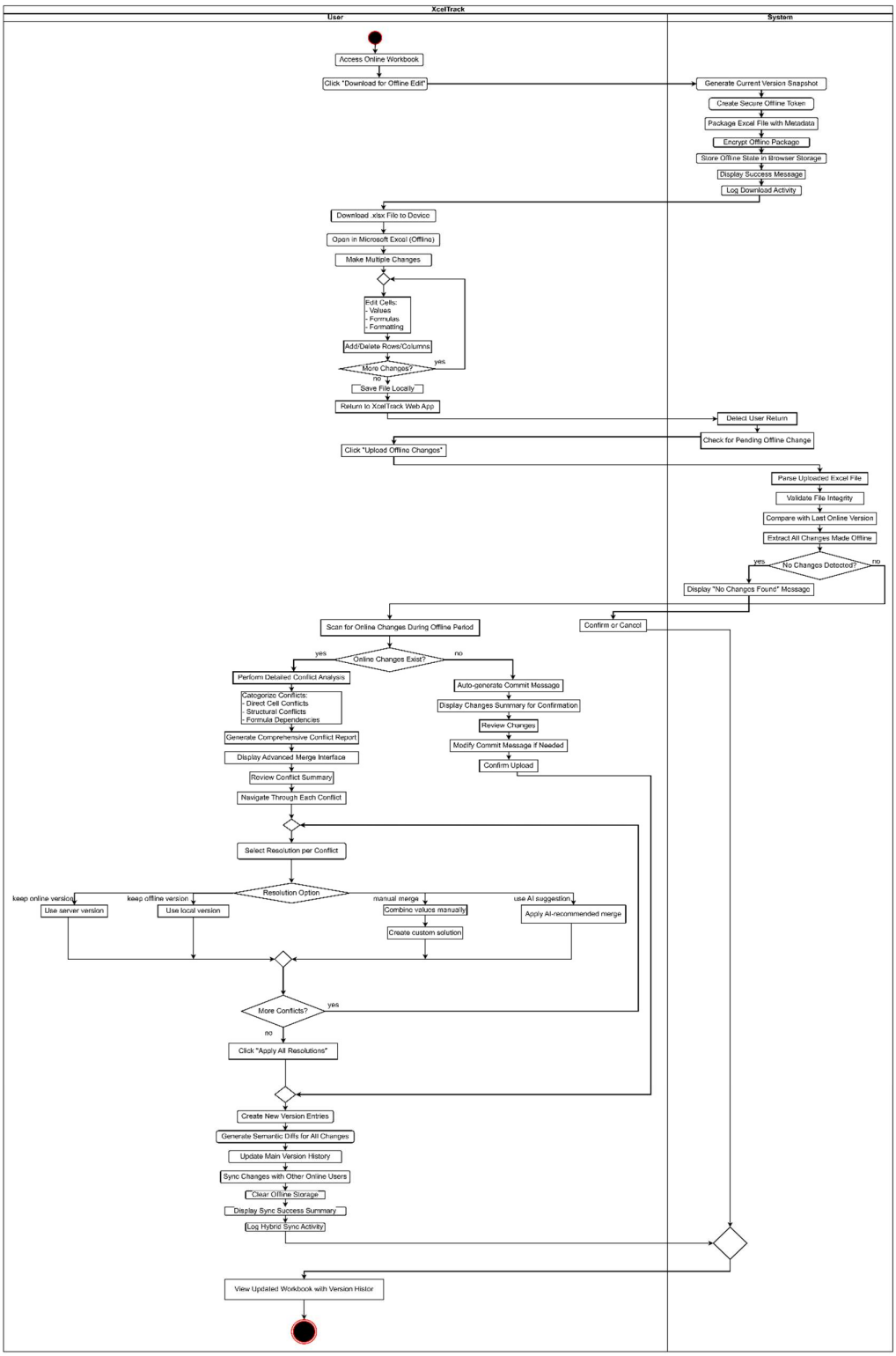

Figure 7: Work in Hybrid Mode (Activity Diagram)

4.1.7. Administer System

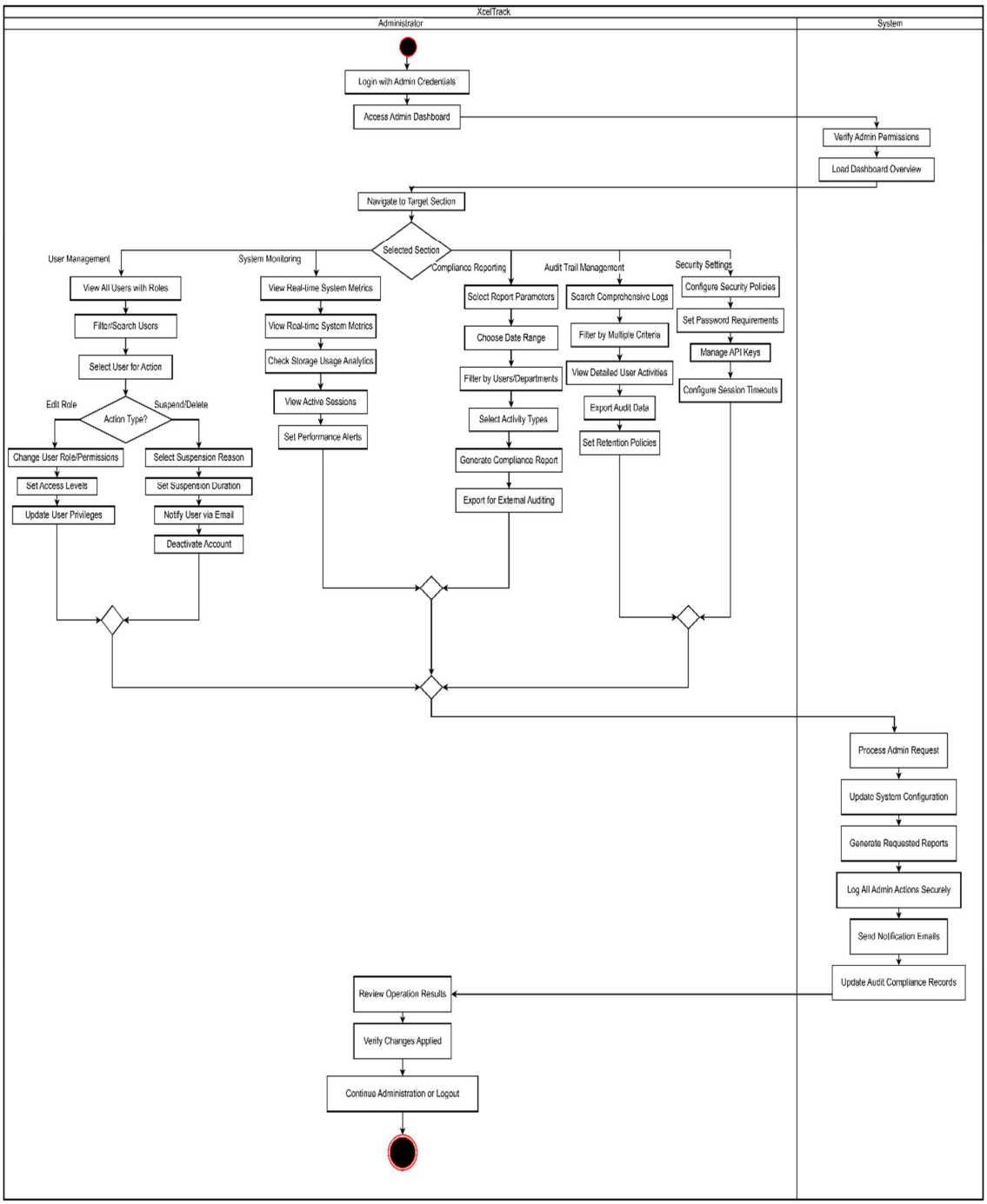

Figure 8: Administer System (Activity Diagram)

4.1.8. Manage Worksheet

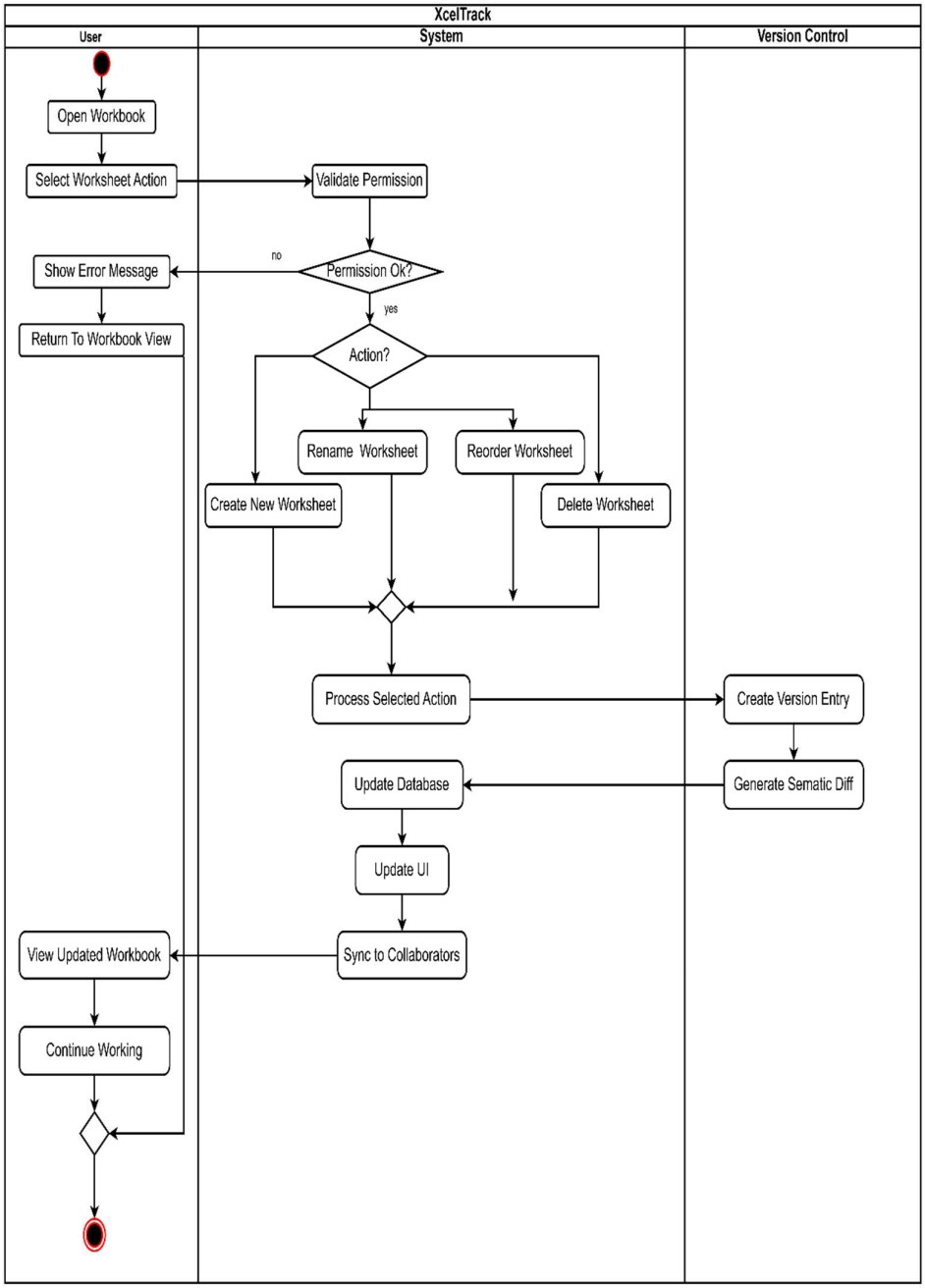

Figure 9: Manage Worksheet (Activity Diagram)

4.2. System Sequence Diagram

4.2.1. User Authentication

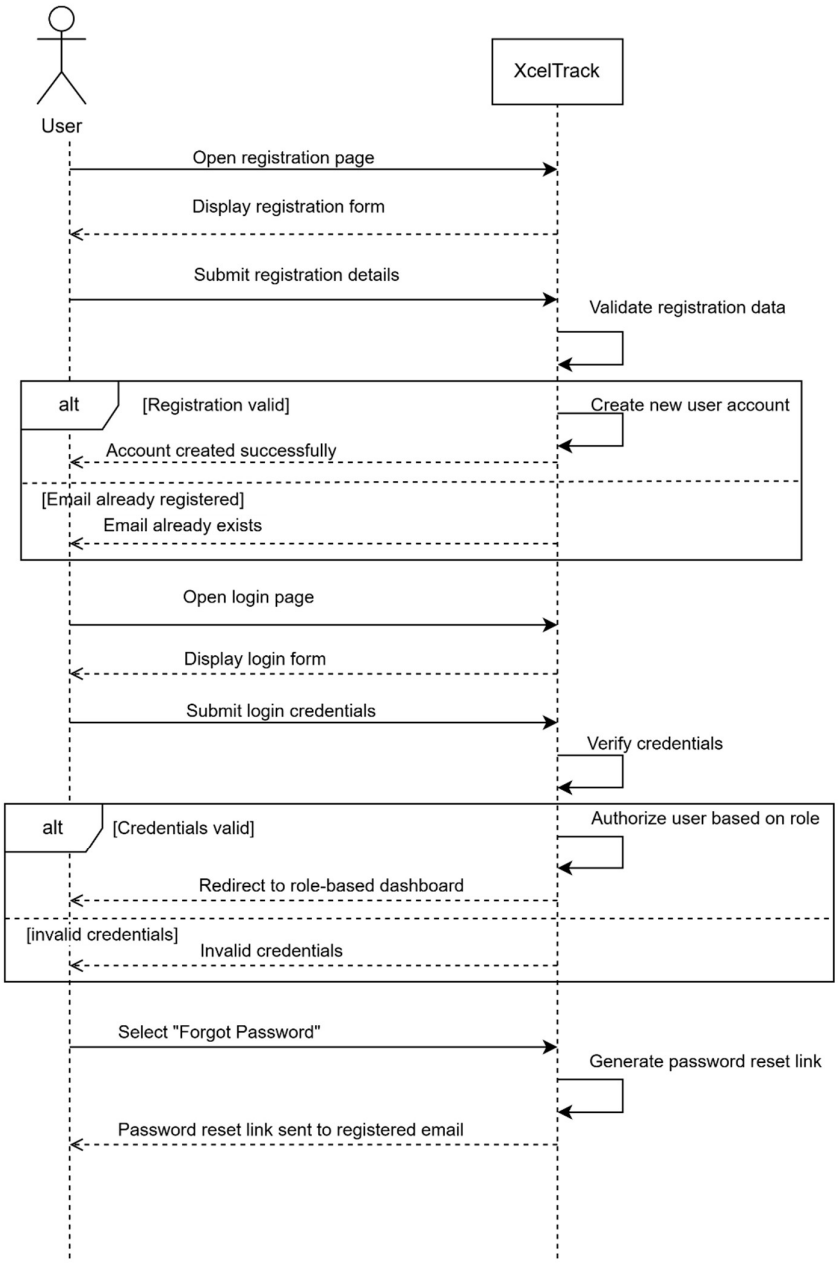

Figure 10: User Authentication (SSD)

4.2.2. Manage Excel Workbooks

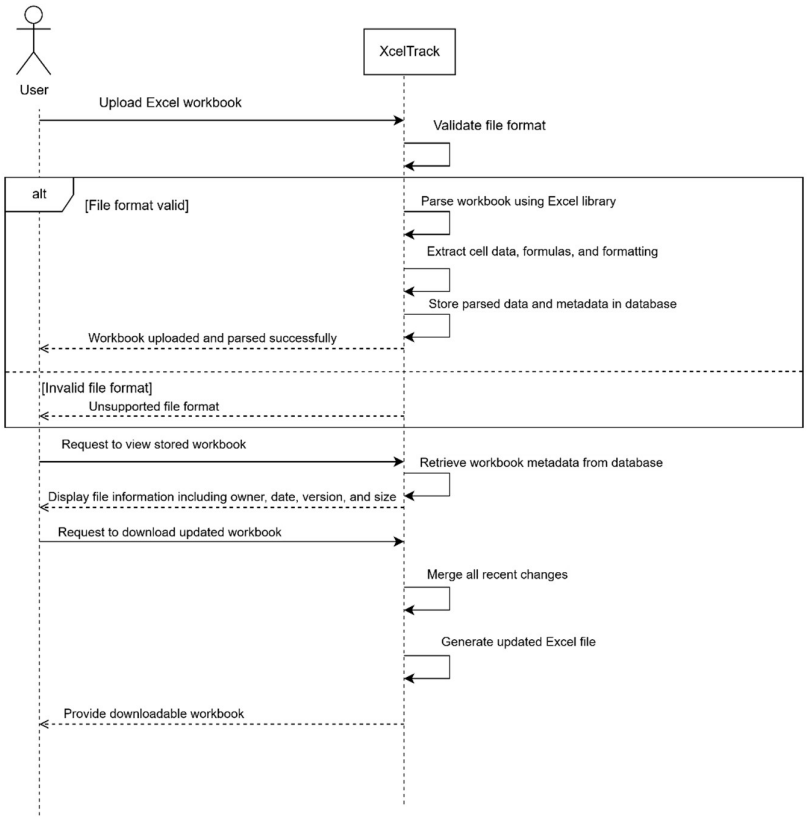

Figure 11: Manage Excel Workbooks (SSD)

4.2.3. Control Workbook Versions

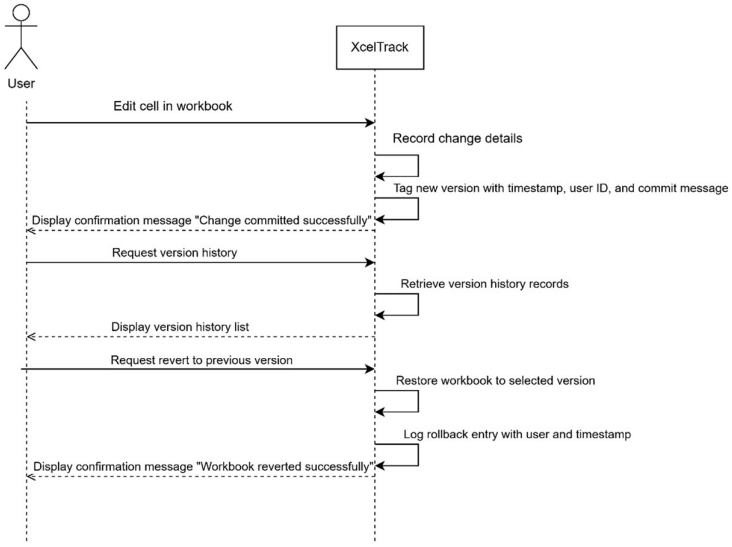

Figure 12: Control Workbook Versions (SSD)

4.2.4. Audit Workbook Changes

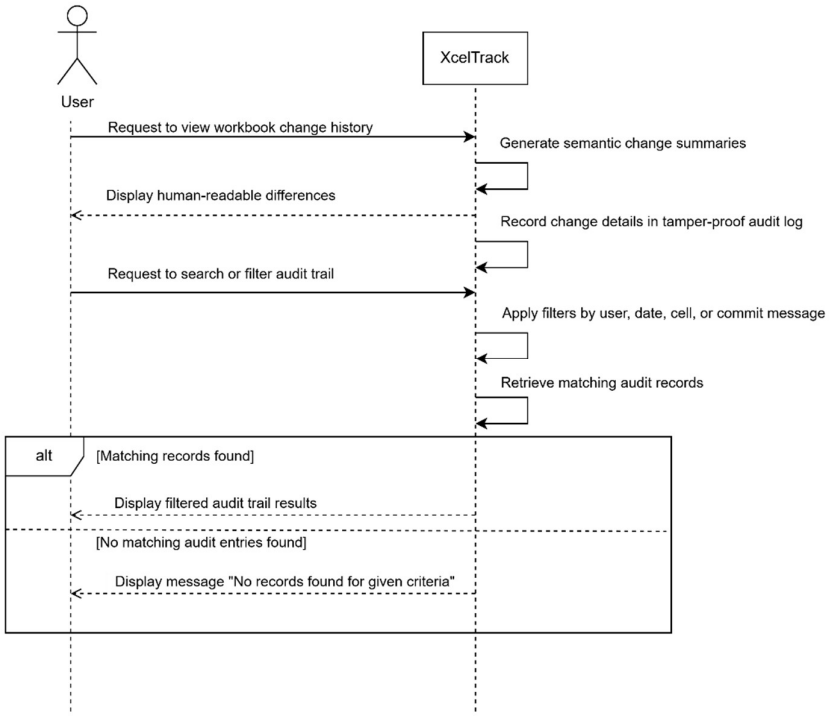

Figure 13: Audit Workbook Changes (SSD)

4.2.5. Get AI Assistance

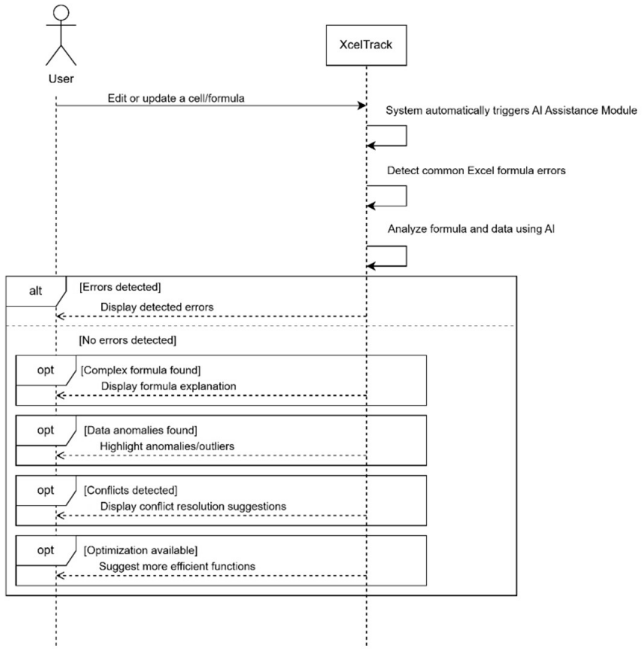

Figure 14: Get AI Assistance (SSD)

4.2.6. Collaborate on Workbooks

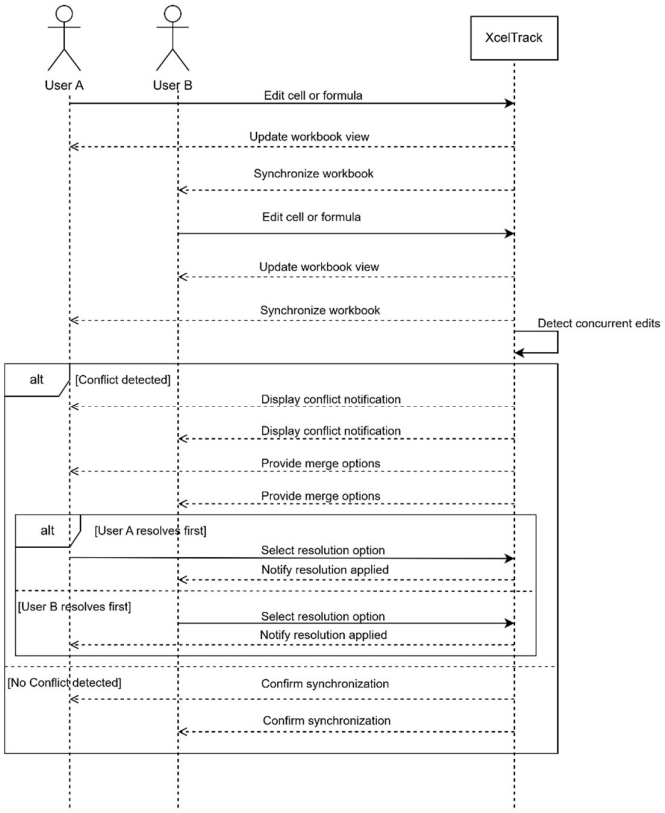

Figure 15: Collaborate on Workbooks (SSD)

4.2.7. Work in Hybrid Mode

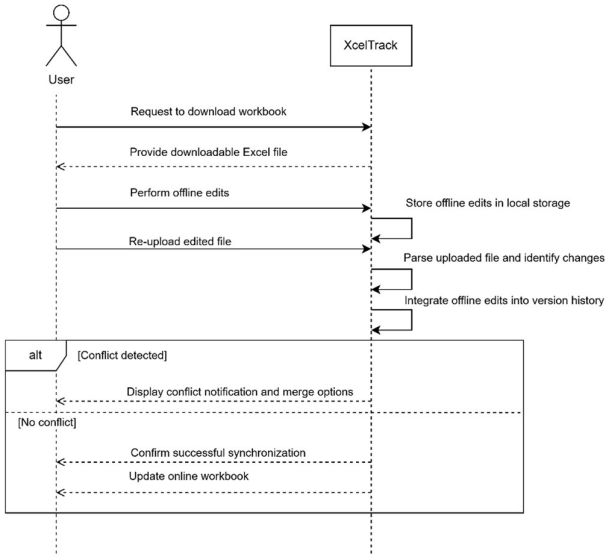

Figure 16: Work in Hybrid Mode (SSD)

4.2.8. Administer System

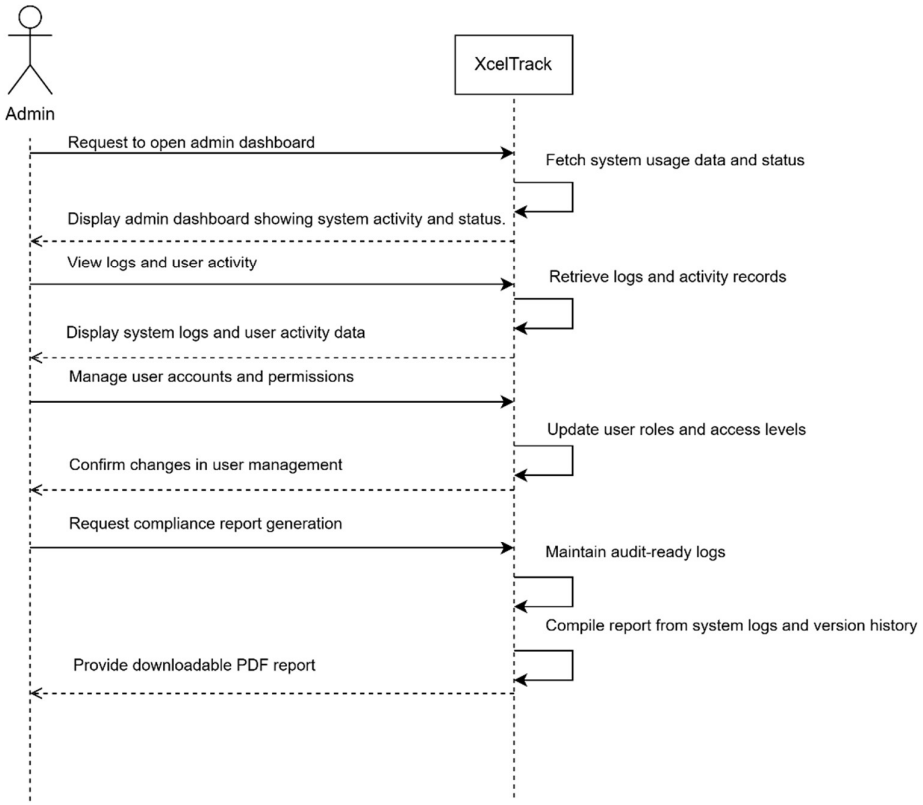

Figure 17: Administer System (SSD)

4.2.9. Manage Worksheet

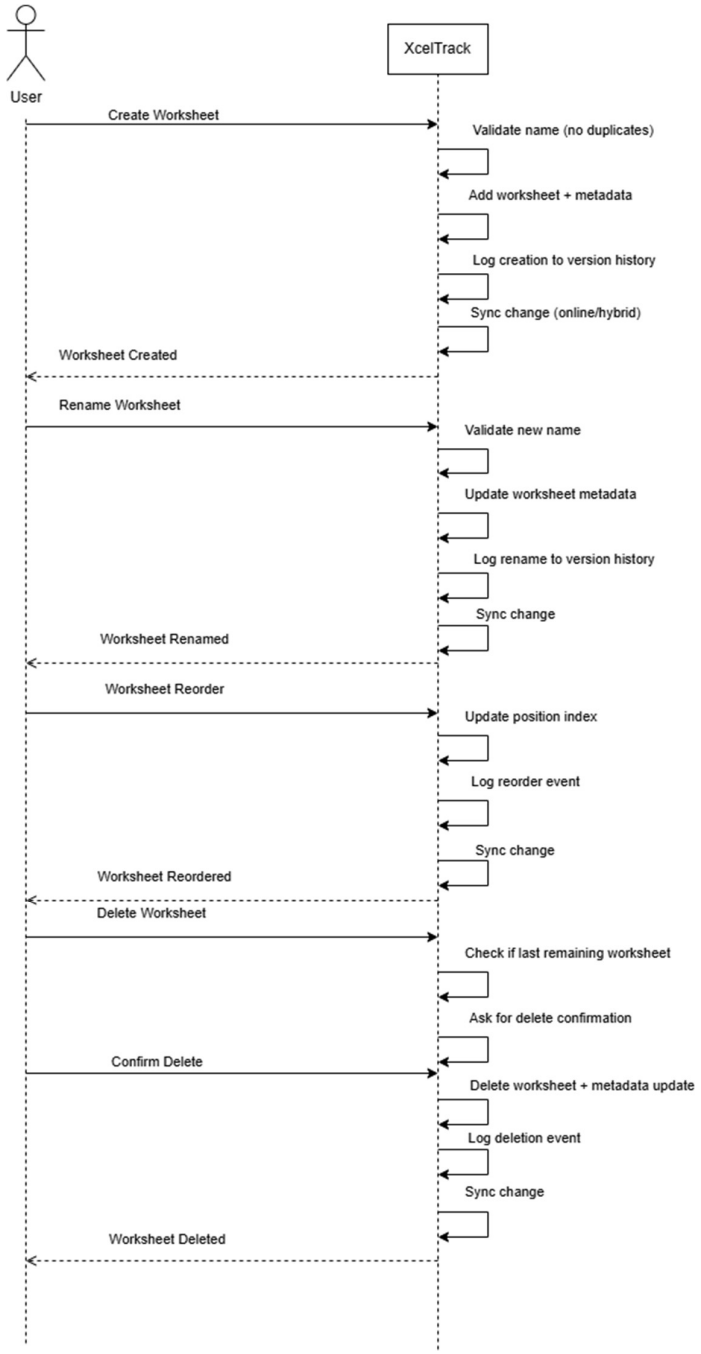

Figure 18: Manage Worksheet (SSD)

4.3. Domain Model

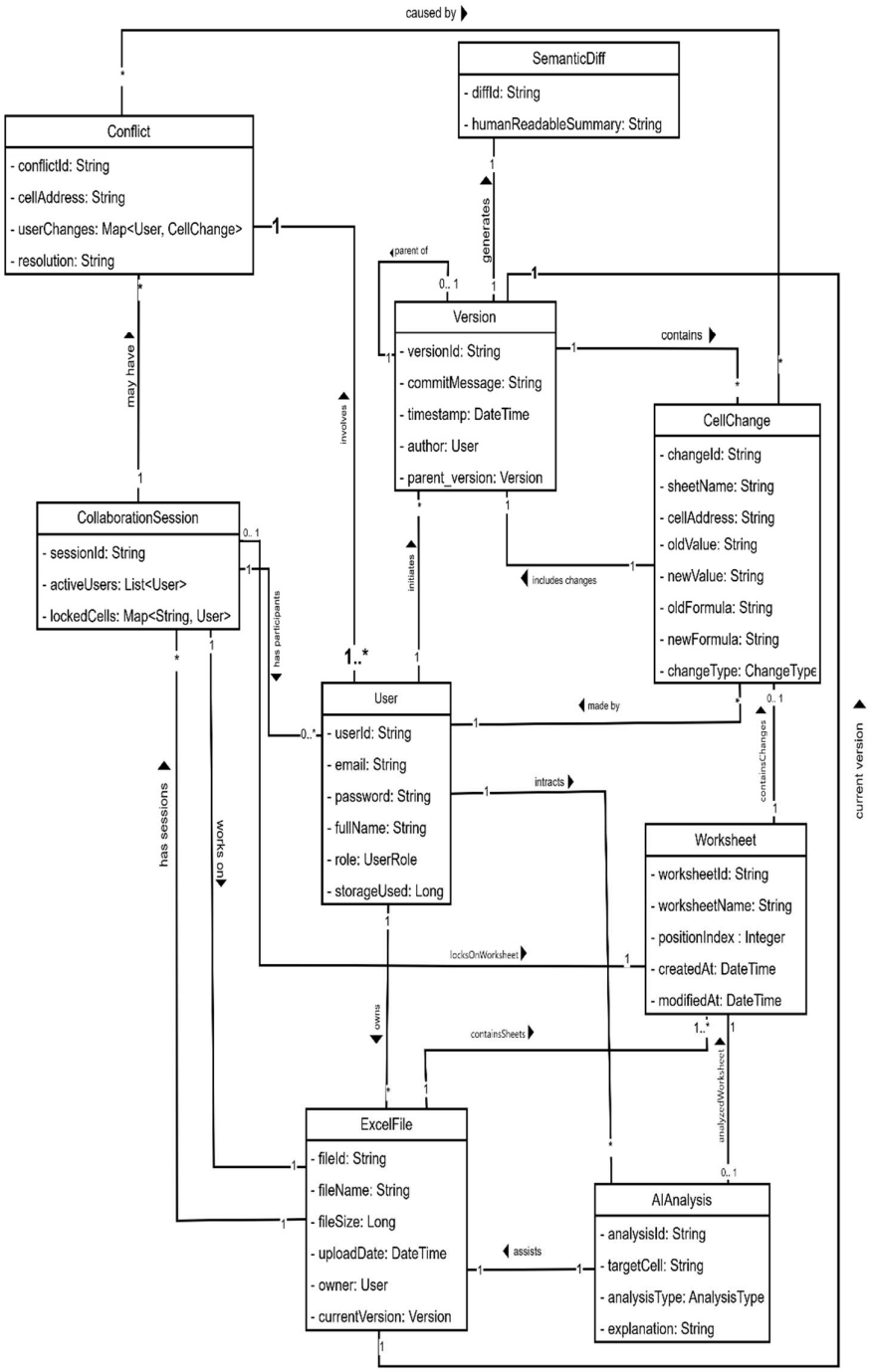

Figure 19: Domain Model Diagram XcelTrack

4.4. Component Diagram

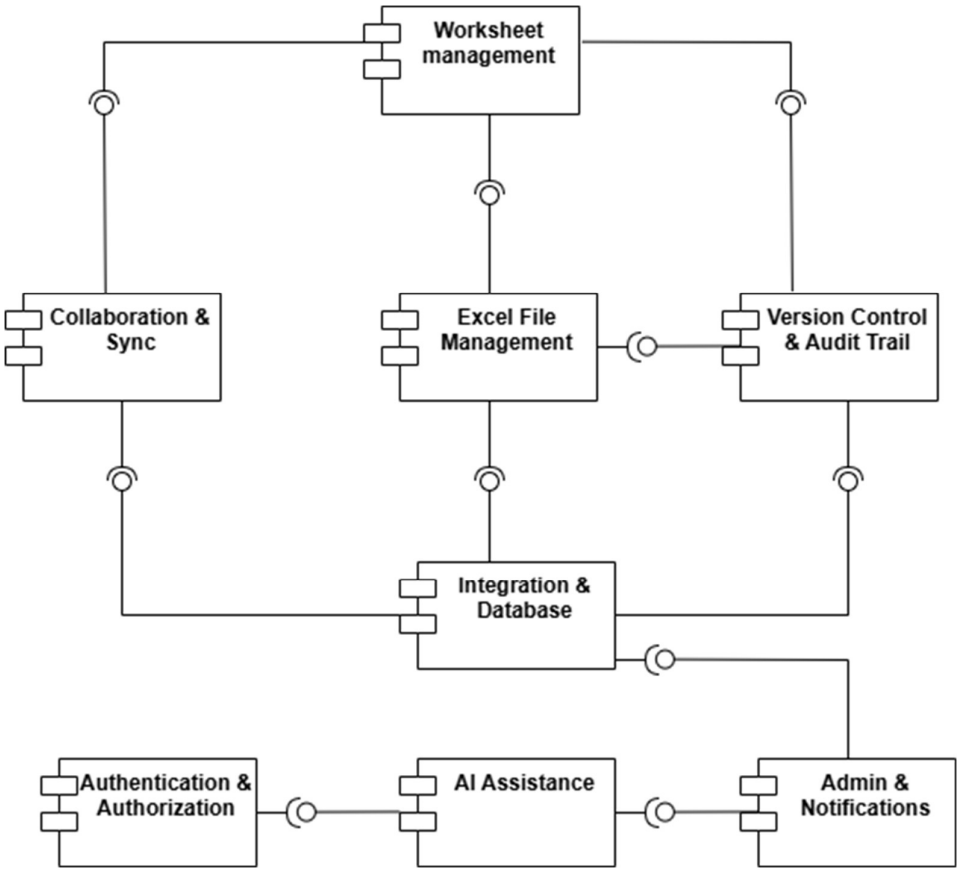

Figure 20: Component Diagram

4.5. Data Flow Diagram

4.5.1. DFD Level-0:

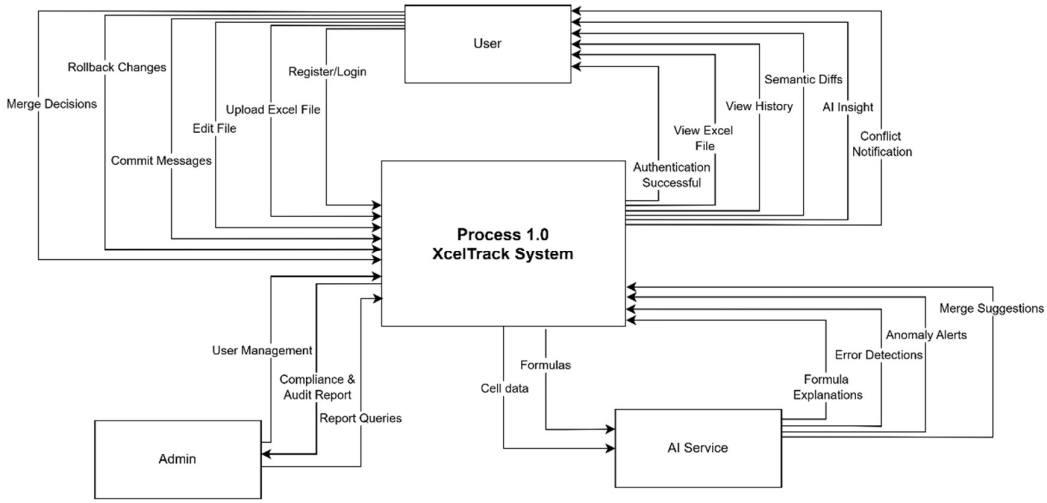

Figure 21: Data Flow Diagram - Level 0

4.5.2. DFD Level-1

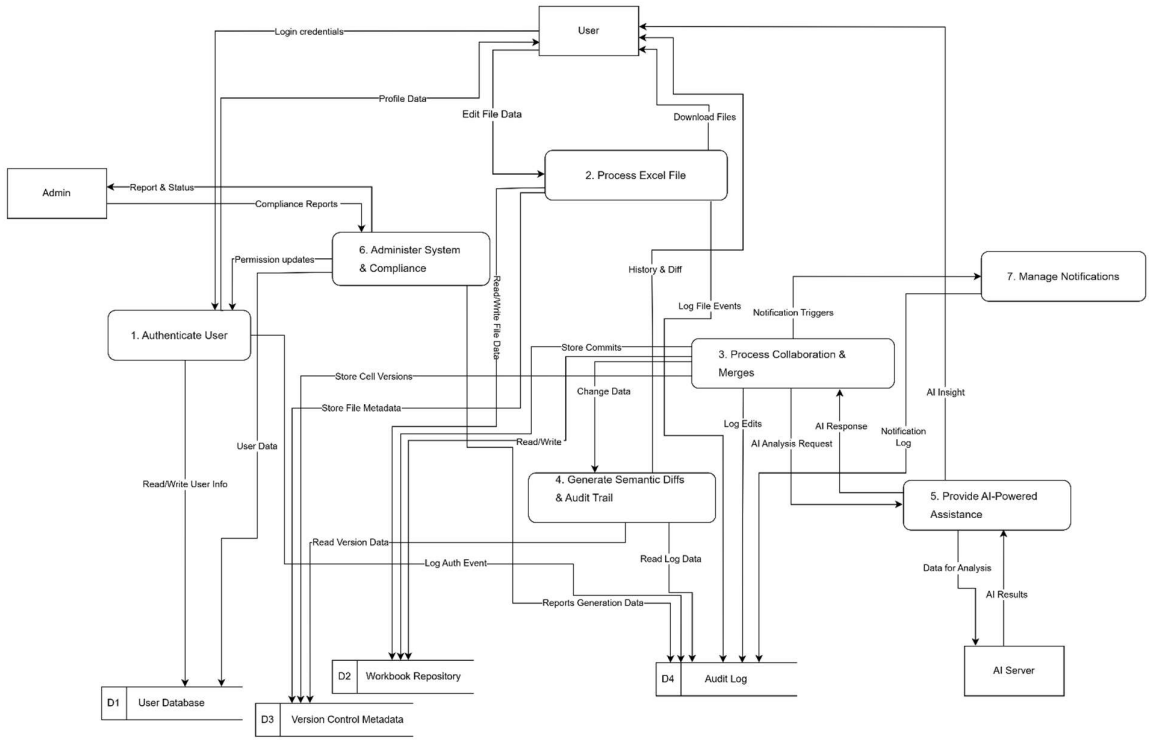

Figure 22: Data Flow Diagram - Level 1

4.5.3. DFD Level-2

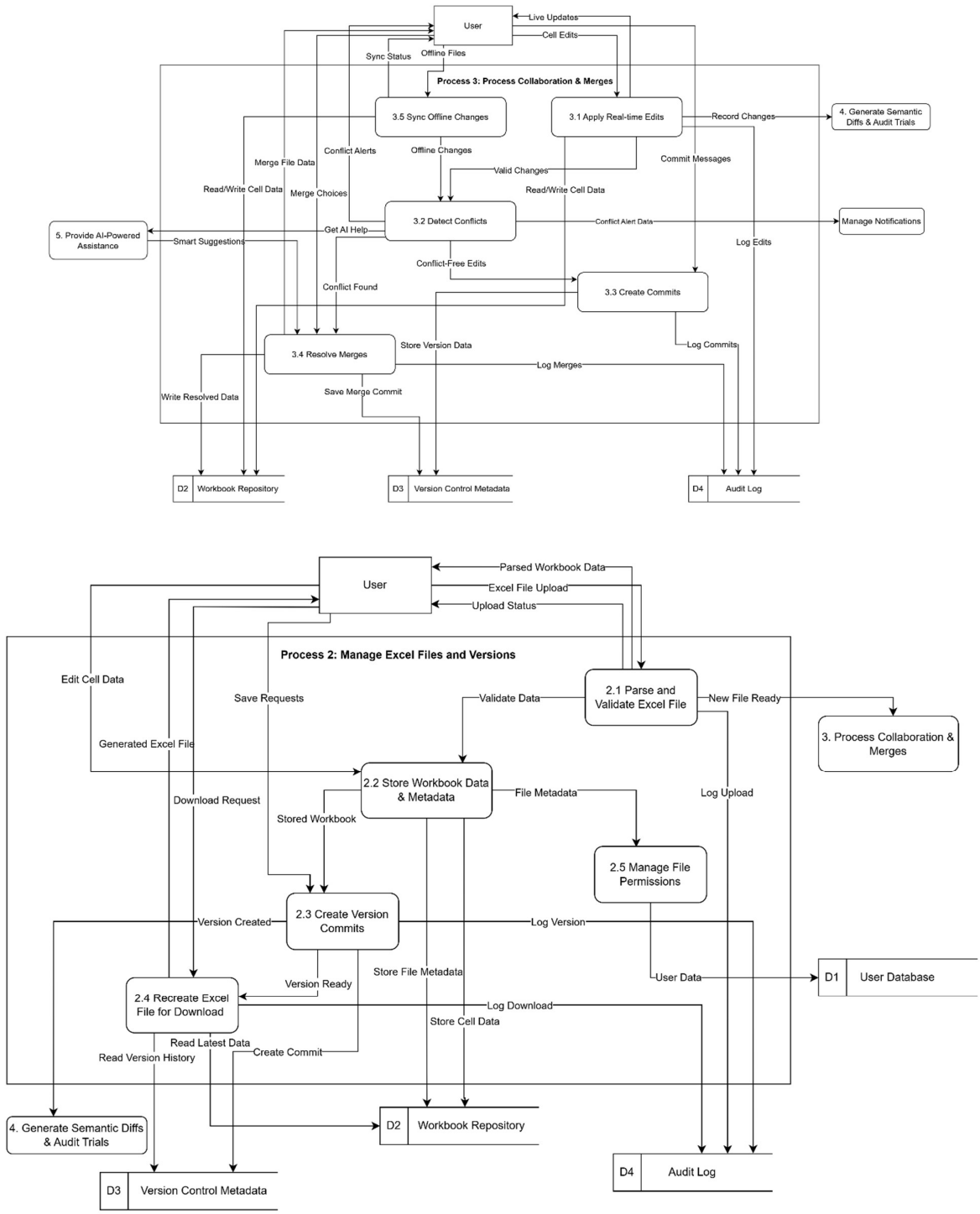

Figure 23: Data Flow Diagram - Level 2

4.6. Entity Relationship Diagram

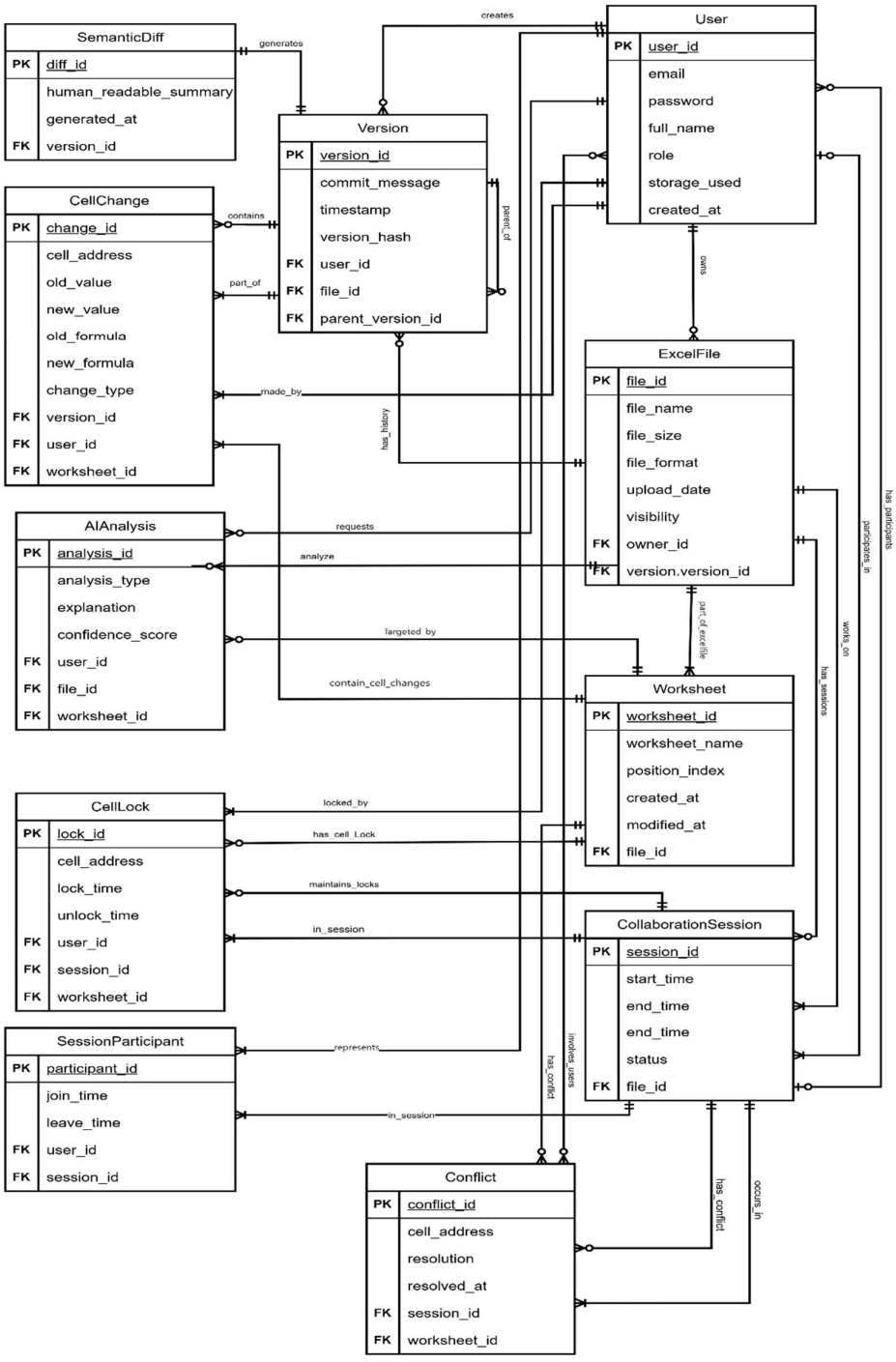

Figure 24: ERD Diagram

4.9. Package Diagram

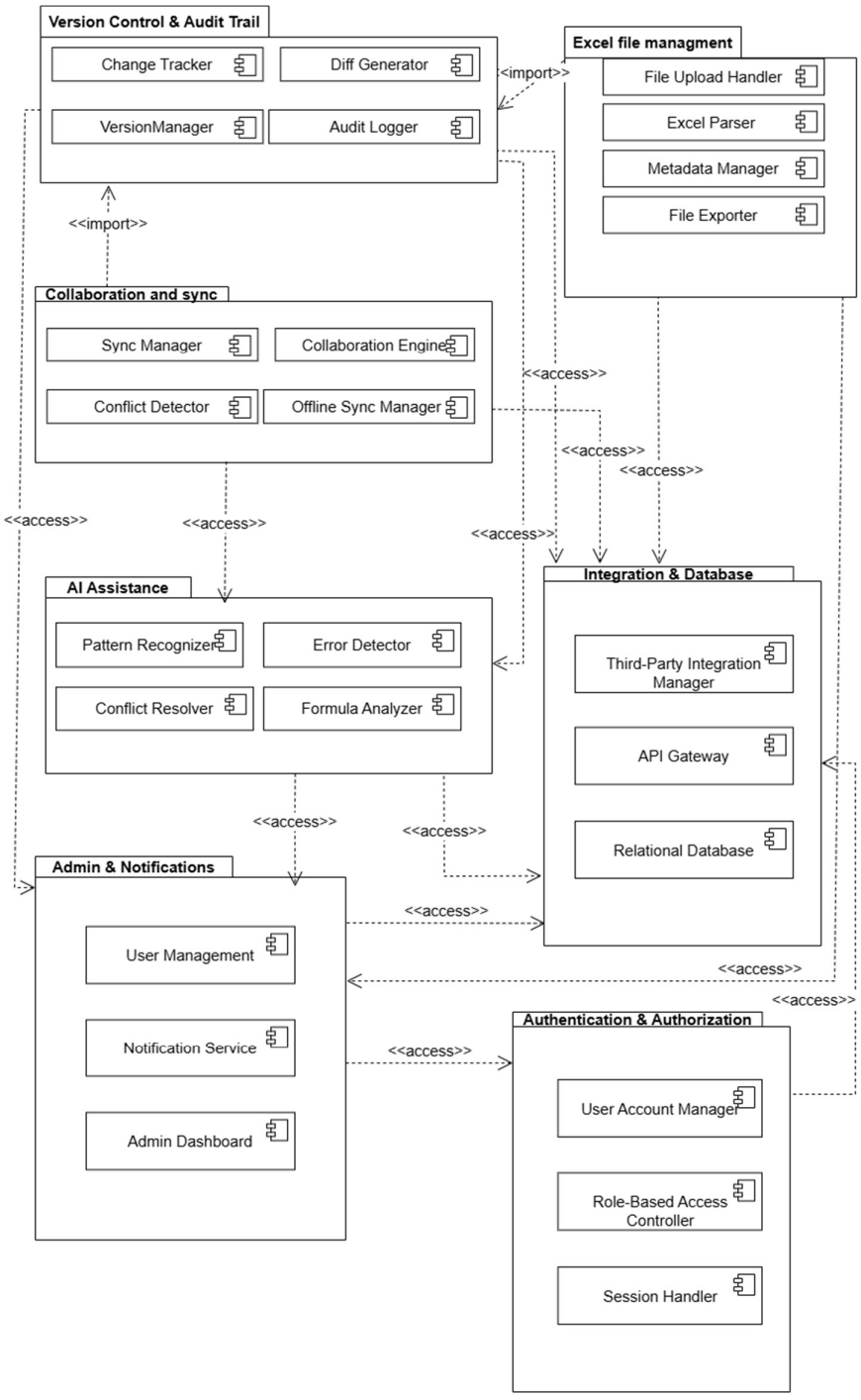

Figure 25: Package Diagram