PW25 - SE Software Engineering 01

System Requirement Specification (SRS)

System: AM Group's ERP

Team Name: Terra Code

Team Members:

1. Ezra Runnath (Team Leader)

2. Fatih Selim Salihoglu

3. Shukurolloh Abdulbokiev

4. KB Monjardin

Stakeholder(s):

1. AM Group GmbH's owners

2. Administrators

3. Managers

4. Property Agents

5. Accountants

1. Introduction

1.1. Purpose

This System Requirements Specification (SRS) document defines the complete functional, non-functional and design requirements for AM Group GmbH's Enterprise Resource Planning (ERP) system. It serves as the main reference for all stakeholders involved in the system's development, validation and deployment.

Primary objectives of this SRS Document are to:

- Provide a clear definition of the system's intended behavior, capabilities and constraints.
- Guide the design, implementation and integration of all system components.
- Define the criteria for system verification, validation and acceptance.
- Serve as a foundation for project planning and risk management.

By documenting requirements in a structured and traceable manner Terra Code ensures with this SRS that the final product aligns with AM Group's operational needs and strategic objectives.

1.2 Scope

The AM Group ERP system is a central, web-based platform designed to bring together and automate the core tasks of running a real estate business. The goal is to replace the current mix of different tools and spreadsheets with one single system where everything is connected.

In Scope Functionalities:

- System Administration & Access Control: Tools for managers to create user accounts, set what each person can see and do, and monitor overall system health and security.
- Property Management: A complete
- Financial Management: A full accounting place for recording all money movement, income, expenses. It tracks payments and creates clear, reliable financial records and reports.
- Manager Dashboard & Analytics: A central screen for owners and managers to see how the business is performing. It combines data into charts and reports and tracks key goals (KPIs).

Out Of Scope Functionalities:

- Handle any tasks outside of the four core modules listed above.

In short, this project delivers one unified system to manage users, properties, finances, and reports. This will make daily work faster and more secure, give managers better insights, and solve the main problems caused by using disconnected tools.

1.3 Definitions, Acronyms and Abbreviations:

No.	Acronyms/Abbreviations	Explanation
1	**ERP**	Enterprise Resource Planning
2	**RBAC**	Role-Based Access Control
3	**GDPR**	General Data Protection Regulation
4	**API**	Application Programming Interface
5	**CRUD**	Create, Read, Update, Delete
6	**KPI**	Key Performance Indicator
7	**SRS**	System Requirements Specification
8	**HTTPS**	Hypertext Transfer Protocol Secure
9	**TLS**	Transport Layer Security
10	**WCAG**	Web Content Accessibility Guidelines
11	**UML**	Unified Modeling Language

1.4 Overview

This System Requirements Specification (SRS) document is organized to provide a clear and structured description of the AM Group ERP system from both functional and non-functional perspectives. Following this introductory section, the document presents the Specific Requirements, which detail how the system interacts with users and external systems, including user interfaces, hardware, software, and communication interfaces. These sections establish the technical environment and boundaries within which the system operates.

Subsequent sections describe the system's functional features through detailed use cases covering system administration, property management, financial management, and data analytics. These use cases define the expected behavior of the system, the roles involved, normal and alternative flows, and exception handling. Supporting design artifacts such as class descriptions, system activities, state machine diagrams, and sequence diagrams are included to clarify system structure and interactions.

The later part of the document focuses on non-functional requirements, including performance, security, usability, reliability, maintainability, portability, and legal

compliance. These requirements define the quality attributes that the system must satisfy to operate effectively in a real-world business environment. The document also outlines key design constraints and the overall software testing strategy to ensure that the system meets its specified requirements before deployment.

Together, these sections provide a complete and traceable specification that serves as a reference for system design, development, testing, and validation, ensuring alignment between Terra Code's implementation and AM Group GmbH's operational and strategic objectives.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

- Compatible web design for both computers and mobile devices.
- Different function access is different.

Describe how the system will interact with its users.

2.1.2 Hardware Interfaces

The AM Group ERP is a web-based application designed to operate independently of specific hardware architectures, relying instead on standard web browser environments.

However, the system requires the following logical hardware interfaces to function correctly on client and server sides:

Client-Side Interfaces (User Devices): The software's primary interaction is through standard web browsers on user hardware.

Characteristic	Specification	How are they supported?
Supported Devices	Desktop Computers & Mobile Devices (Smartphones/Tablets).	Support is achieved through **responsive design** ensuring the UI adapts dynamically to all screen sizes and orientations.

Display Interface	Functional display adapter capable of rendering modern web standards (HTML5/CSS3).	Logical support for **full-screen, graphical presentation** (as opposed to line-by-line terminal support). The interface requires **high contrast and text scaling** to support accessibility standards (WCAG 2.1 AA).
Input Interface	Standard Human Interface Devices (HIDs) such as Keyboard, Mouse, and Touchscreen.	The system relies on standard browser APIs to process input events, supporting both **precise cursor control** (desktop) and **gesture-based input** (mobile).
Network Interface (NIC)	Requires a functional Network Interface Card (Wi-Fi or Ethernet) to establish a TCP/IP connection.	Relies on the host device's operating system to manage the connection, utilizing **standard DNS resolution** for locating the ERP domain.

Server-Side Interfaces (Hosting Environment): The server hardware must offer specific instruction sets and configuration to securely and efficiently host the application.

Characteristic	Specification	How are they supported?
Instruction Sets	Virtual or Physical **x86-64** Architecture (Standard for hosting environments).	The **software's compiled environmen**t (e.g., PHP Runtime, Database Engine) must interface with **a 64-bit instruction set** for **efficiency** and **memory** addressing (as required by performance efficiency NFR [1]).
Storage Interface	Logical interface to persistent, high-speed **solid-state drives** with expansion capability.	Must support storage for **at least 10,000 property records** and associated image assets. The interface must support I/O operations necessary for daily database backups [1].
Database Interface	**Internal logical network** interface to communicate between the **Application Server** and the **Database Server**.	This interface is critical for **concurrent data CRUD operations** without data loss or inconsistency, and for ensuring real-time data aggregation for the **Manager Dashboard** [1].

Logical Configurations (Protocols): To facilitate secure communication, the server hardware interface must be configured to listen on specific logical ports and protocols.

Port	Protocol	Purpose in AM Group ERP (Protocol Use)
443	HTTPS (TLS/SSL)	**Primary Interface**: Encrypts user credentials and data using **AES-256** or **SHA-256**. Ensures **GDPR compliance** during transmission [1].
80	HTTP	**Redirection Only**: Used to enforce security by immediately redirecting all insecure traffic to **HTTPS** (**Port 443**). SSL.
3306 (or similar)	SQL Protocol	**Internal Interface**: Protocol used exclusively for the **Application Server** to communicate with the **Database Server**. **Security** is maintained by disallowing public access to this port.
22	SSH	**Admin Interface**: Protocol used by Administrators to perform secure system **maintenance**, **server configuration**, and **manual data recovery** operations.

2.1.3 Software Interfaces

The software components and external systems that the AM Group ERP must interface with in order to operate correctly follows a layered web application architecture consisting of a client layer, application layer, and data layer. All software interfaces are designed to support scalability, security, and maintainability.

The client-side interface operates through standard web browsers and does not require additional software installation.

Software Component	Specification	Interface Description
Web browser	Google Chrome, Mozilla Firefox, Microsoft Edge, Safari	Provides access to the ERP system via HTTPS. The UI is rendered using HTML5, CSS3, and JavaScript.
Operating system	Windows, macOS, Linux, Android, iOS	Browser-based access ensures OS independence.
Frontend Framework	JavaScript-based framework (e.g., React or equivalent)	Communicates with backend services using RESTful APIs and JSON payloads.

Server-side Software Interfaces

The server-side software interface of the AM Group ERP system consists of the web server, application server, database management system, and supporting internal services that

together deliver the system's core functionality. A web server (such as Apache or Nginx) is responsible for handling incoming HTTP/HTTPS requests and forwarding them to the application layer.

The application server executes business logic, enforces role-based access control (RBAC), manages workflows, and processes user requests originating from the client interface. Persistent data is managed through a relational database management system (such as MySQL, MariaDB, or PostgreSQL), which stores user accounts, property records, financial transactions, system logs, and analytical data.

Communication between the application server and database server occurs through secure internal connections to ensure data consistency, integrity, and performance. Additionally, internal authentication and analytics services interface with the application layer to manage secure user sessions, audit logging, KPI calculations, and report generation.

External Software Interfaces

The AM Group ERP system interfaces with several external software services to support non-core but essential functionalities. An external email service, accessed through the Simple Mail Transfer Protocol (SMTP), is used to send automated notifications such as password reset links, system alerts, and administrative messages. Backup and recovery software interfaces with the database layer to perform scheduled backups and data restoration operations, ensuring business continuity and data protection.

The system also integrates with external logging and monitoring tools to capture system events, errors, and user activity for auditing, performance monitoring, and compliance purposes. All external software interfaces are accessed through secure and standardized protocols, ensuring interoperability while maintaining the confidentiality and integrity of system data.

2.1.4 Communication Interfaces

The AM Group ERP system uses secure and standardized communication interfaces to support reliable interaction between users, internal system components, and administrative services. These interfaces ensure data confidentiality, integrity, and availability while supporting concurrent multi-user access in a web-based environment.

Client-Server Communication

Communication between end users and the ERP system is conducted over standard internet protocols. All data exchanged between client devices and the application server is transmitted using encrypted channels to protect sensitive business and personal information.

- **HTTPS (TLS/SSL):**
  Used as the primary communication protocol between client browsers and the ERP system. It encrypts all transmitted data, including login credentials and transactional information, to comply with GDPR and security requirements.
- **HTTP (Redirection Only):**
  Used exclusively to redirect all unsecured traffic to HTTPS, ensuring that no sensitive data is transmitted over insecure connections.

- **TCP/IP:**
  Serves as the underlying transport protocol to provide reliable and ordered data delivery between clients and servers.

API-Based Communication

The system employs RESTful Application Programming Interfaces (APIs) to enable communication between the frontend and backend components. These APIs ensure modularity, scalability, and interoperability.

- **API Architecture:** REST-based
- **Data Format:** JSON
- **Authentication:** Session-based or token-based authentication
- **Access Control:** Role-Based Access Control (RBAC) enforced at the API level

Each API request is validated to confirm the user's identity and role before allowing access to system resources or operations.

Internal System Communication

Internal communication between server-side components, including the application server, database server, and analytics modules, occurs over secured internal network interfaces. These communication channels support high-frequency data exchanges such as CRUD operations, report generation, and real-time dashboard updates.

- Database communication is restricted to trusted services within the hosting environment.
- Concurrent data access is managed to prevent data loss or inconsistency.
- Internal interfaces support transactional integrity and audit logging.
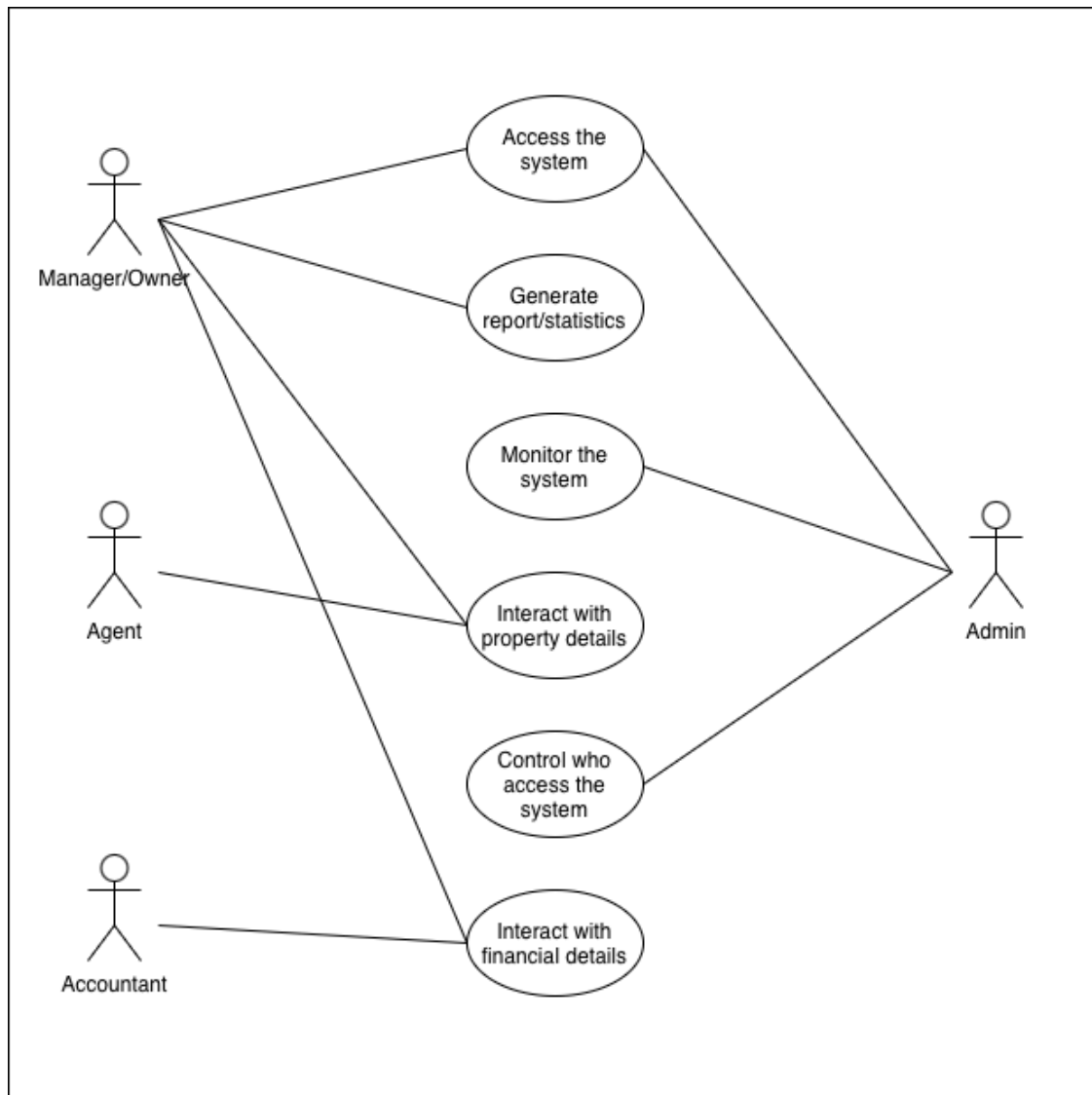
Administrative Communication

Administrative and maintenance-related communication is handled through secure channels to prevent unauthorized access to system infrastructure.

- **SSH:**
  Used by system administrators for secure remote access to the server for configuration, updates, monitoring, and recovery operations.
- **Secure Database Connections:**
  Prevents public access to database services and restricts access to authorized internal components only.

2.2. System Features

This section outlines the key use cases that define the functional behavior of the system. Each use case describes how the system interacts with its primary actors to achieve specific objectives essential to business operations. The use cases are organized according to the main functional modules of the system, namely System Administration, Property Management, and Financial Management. These descriptions provide a high-level understanding of the system's intended functionalities, user interactions, and operational workflows, forming the foundation for the detailed requirements specified in later sections of the SRS.

The System Administration module allows the Administrator to manage and maintain the overall system environment. This includes setting up user accounts, configuring system parameters, monitoring system activity, and resolving operational issues. The goal of this use case is to ensure that the system remains secure, stable, and accessible to authorized users. All administrative actions help support the seamless functioning of other modules such as property management and financial management.

The Property Management module enables the property agent to maintain information related to properties within the system while the manager oversees those actions. This includes adding new properties, updating existing property details, removing outdated records, and viewing or searching through the property database. The manager is also able to perform all the actions that are available to the property agent. The primary objective of this use case is to ensure accurate, up-to-date, and easily accessible property information to support business operations.

The Financial Management module allows the accountant to record and manage all financial transactions related to the business. This includes documenting payments received, recording

expenses, updating or deleting financial records, tracking outstanding payments, and generating financial reports. Much like the interaction of the manager with the property management module, they are also able to access the features available to the accountant in this module as well. The purpose of this use case is to maintain clear, accurate, and auditable financial data.

The Data Analytics module enables the Manager to generate insights from the system's stored data to support informed decision-making. This includes selecting relevant datasets, initiating analytical processes, viewing generated reports, and interpreting the results to guide operational or strategic actions. The module provides automated data processing and analysis capabilities, allowing users to identify trends, monitor performance, and evaluate key metrics related to properties, finances, and overall business activities. The goal of this use case is to deliver accurate, timely, and meaningful analytical outputs that enhance managerial effectiveness and improve organizational planning.

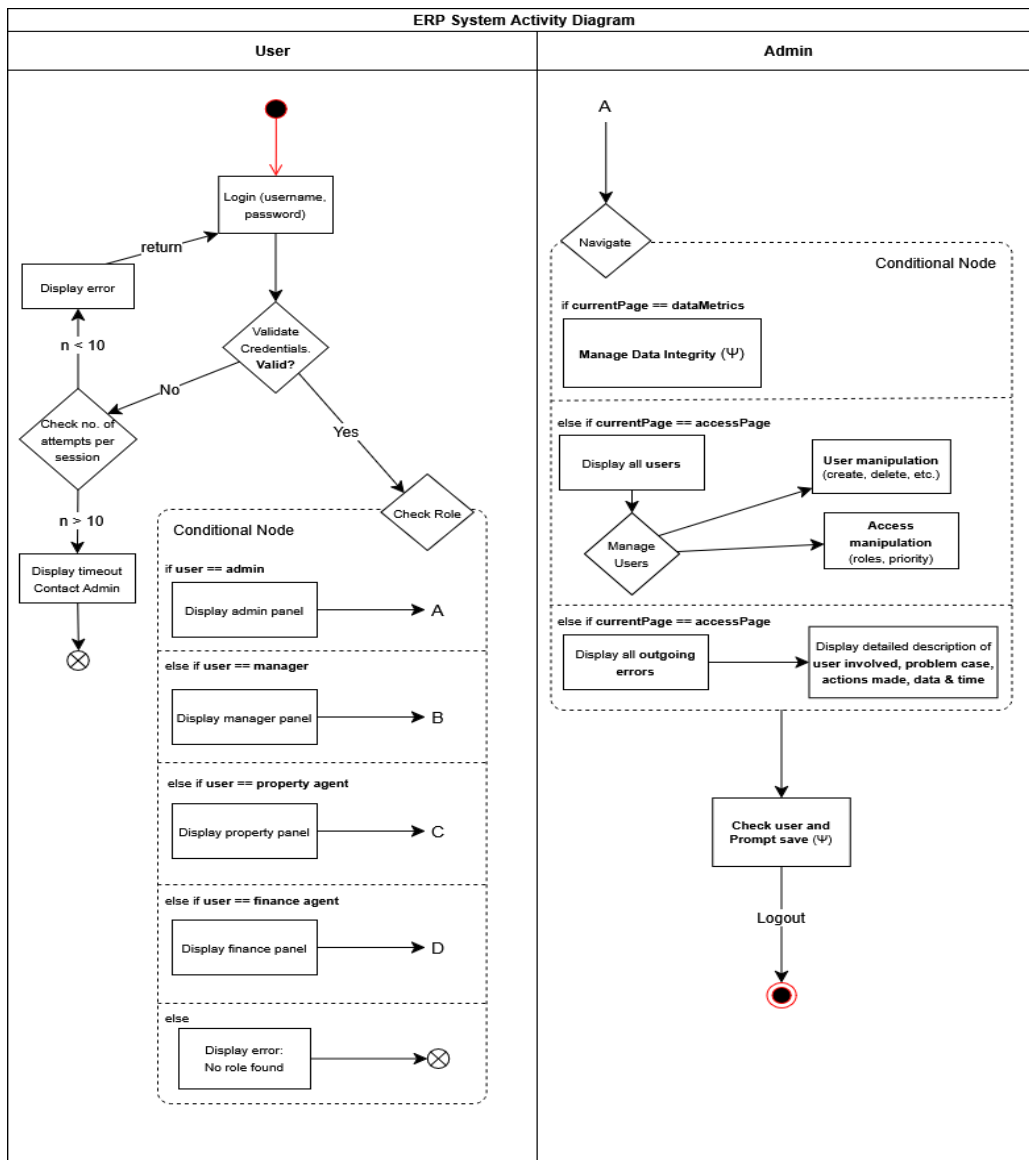## 2.2. System Activities

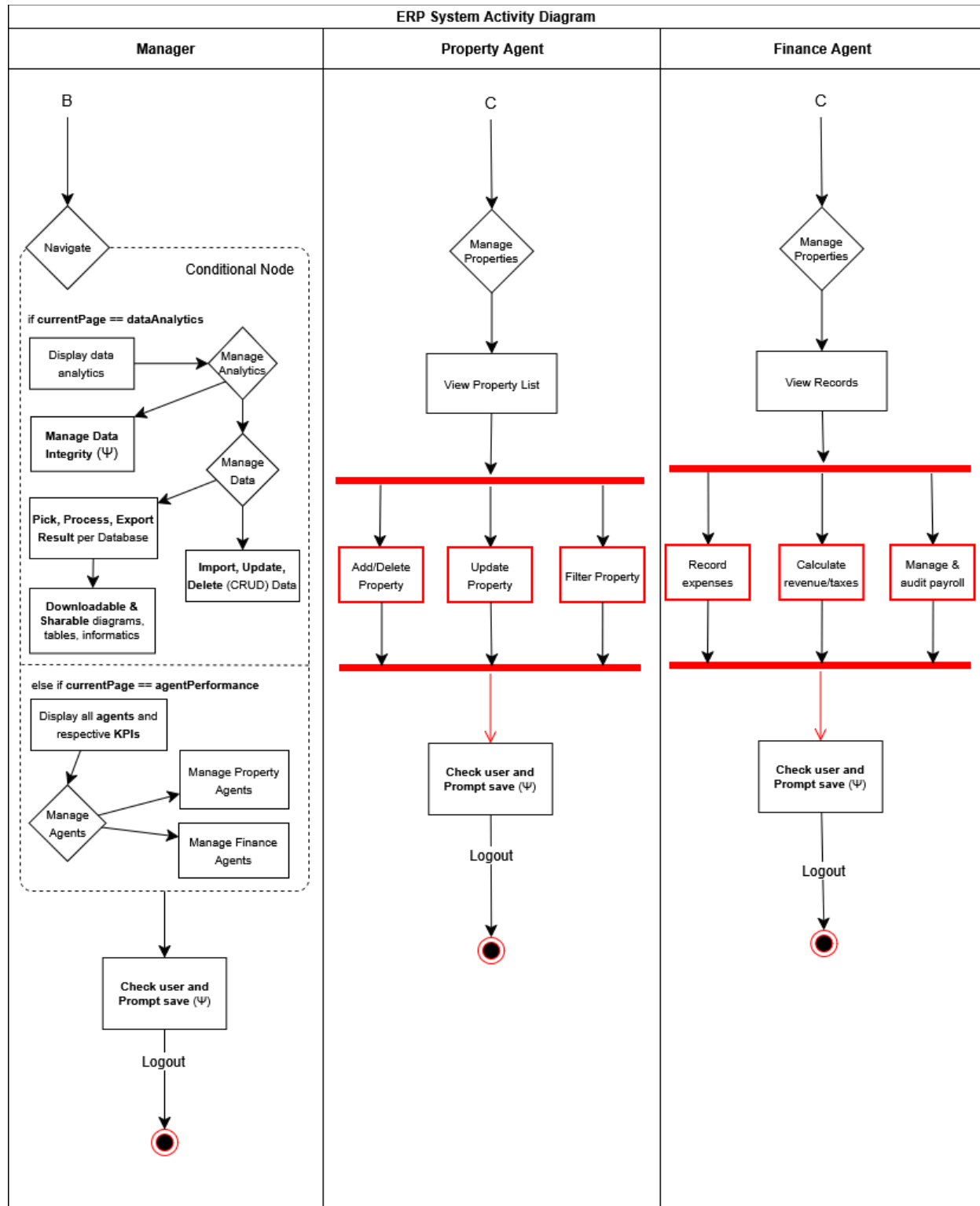Figure 2.2.1: Activity Diagram for Web-based ERP

Figure 2.2.2: Activity Diagram for Web-based ERP

Index of Shared Activities

Activity: Manage Data Integrity

- Display data metrics (e.g. Integrity)
- Manage Metrics
- Verification tools
- Integration tools
- Querying tools for metrics (CRUD)
- Show results

Activity: Check user and Prompt save

- Re-validate Role
- Valid → Display & Prompt "Final changes?" → Save → Write detailed log
- Invalid → Display error: Invalid Permissions → Write detailed log

Figure 2.2.3: Index of Shared Activities

2.3. System Classes

This class diagram represents a property management ERP system designed to support different user roles and core business operations. The system is structured around role-based users (Administrator, Manager, Agent, and Accountant) who interact with business entities such as Property, Customer, FinancialRecord, and Report. Administrative users control system configuration and user management, managers oversee operations and analytics, agents handle property-related tasks, and accountants manage financial records. CRUD functionality is separated into dedicated service classes to clearly define responsibilities and improve maintainability.

Administrator

+ userID: int
+ username: string
+ password: string
+ role: string

+ login()
+ logout()
+ forgotPass()
+ manageUsers()
+ configureSystem()
+ viewLogs()

Report

+ reportID: int
+ reportDate: date
+ flagged: boolean
+ summary: string

Manager

+ userID: int
+ username: string
+ password: string
+ role: string

+ login()
+ logout()
+ forgotPass()
+ propertyCRUD()
+ financeCRUD()
+ runAnalysis()
+ generateGraphs()
+ getKPI()

propertyCRUD()

+ registerProperty()
+ viewProperty()
+ updateProperty()
+ removeProperty()

Agent

+ userID: int
+ username: string
+ password: string
+ role: string

+ login()
+ logout()
+ forgotPass()
+ propertyCRUD()

Accountant

+ userID: int
+ username: string
+ password: string
+ role: string

+ login()
+ logout()
+ forgotPass()
+ financeCRUD()

financeCRUD()

+ registerRecord()
+ viewRecord()
+ updateRecord()
+ removeRecord()

Customer

+ customerID: int
+ name: string
+ contact: string

Property

+ propertyID: int
+ address: string
+ type: string
+ status: string
+ value: decimal

FinancialRecord

+ transactionID: int
+ date: date
+ amount: decimal
+ description: string
+ propertyID: int

2.3.1. Administrator: System-level control

The Administrator is responsible for maintaining and controlling the system at a global level. This role ensures that the system operates securely, efficiently, and according to organizational policies. The Administrator manages user access by creating, updating, and removing user accounts, as well as assigning appropriate roles and permissions. In addition, the Administrator configures system-wide settings to support the correct functioning of all modules. The Administrator is also able to monitor system activity through logs in order to identify errors, track usage, and maintain system integrity. Authentication-related operations such as logging in, logging out, and password recovery are supported to ensure secure access to administrative functionalities.

Attributes:

- userID, username, password, role

Operations:

- login(), logout(), forgotPass()
- manageUsers()
- configureSystem()
- viewLogs()

2.3.2. Manager: Operational and analytical control

The Manager plays a central role in overseeing daily operations and making data-driven decisions. This role has access to both operational and analytical functionalities within the system. The Manager can manage property-related information by performing create, read, update, and delete operations, as well as handle financial records such as payments and expenses.

In addition, the Manager is able to initiate data analysis processes, generate analytical reports and visualizations, and retrieve key performance indicators (KPIs) to evaluate business performance. The Manager is also connected to reports, agents, and accountants, allowing coordination between operational, financial, and analytical activities. Standard authentication features, including login, logout, and password recovery, are provided to ensure secure access.

Attributes:

- userID, username, password, role

Operations:

- login(), logout(), forgotPass()
- propertyCRUD()
- financeCRUD()
- runAnalysis()
- generateGraphs()
- getKPI()

Relationships:

- A Manager is linked to:
  - Reports (data)
  - Agents
  - Accountants

2.3.3. Agent: Property operations

The Agent is responsible for handling property-related operations within the system. This role focuses on managing property information, including adding new properties, updating existing

records, and removing outdated entries. Agents interact with the system primarily to support property listings and updates, ensuring that property data remains accurate and current. Authentication functionalities such as login, logout, and password recovery are available to allow secure access to the property management features assigned to this role.

Attributes:

- userID, username, password, role

Operations:

- login(), logout(), forgotPass()
- propertyCRUD()

2.3.4. Accountant: Financial management

The Accountant is responsible for managing the financial aspects of the system. This role handles financial records by performing create, read, update, and delete operations on payments and expenses. The Accountant ensures that financial data is accurately recorded and maintained to support reporting, auditing, and decision-making processes. Secure authentication mechanisms, including login, logout, and password recovery, are provided to control access to financial management functionalities.

Attributes:

- userID, username, password, role

Operations:

- login(), logout(), forgotPass()
- financeCRUD()

2.3.5. Core Business Entities

Property: The Property class represents real estate assets managed by the system. Each property has attributes such as address, type, status, and value. Properties are managed by Agents, linked to customers, and associated with multiple financial records.

Customer: The Customer class stores information about clients involved in property transactions. Customers can be associated with multiple properties, allowing the system to track ownership or rental relationships.

FinancialRecord: The FinancialRecord class represents individual financial transactions related to properties. Each record includes transaction details such as date, amount, and description and is linked to a specific property. These records are managed by Accountants.
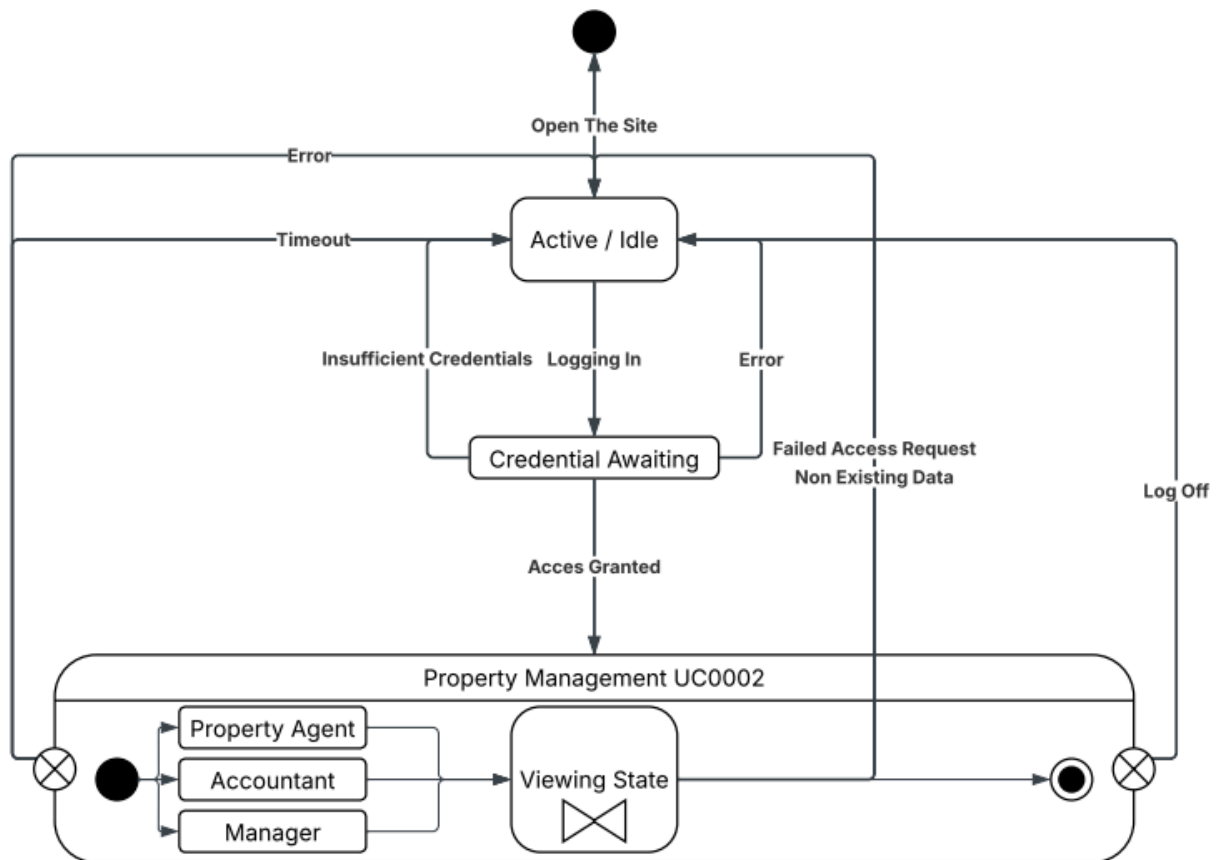
Report: The Report class represents analytical and operational summaries generated by Managers. Reports include a date, summary, and a flagging status and are used to support monitoring and decision-making.

PropertyCRUD: The propertyCRUD class provides operations for managing property data, including registering, viewing, updating, and removing properties. It supports the functionality used by Agents and Managers without embedding CRUD logic directly into the Property class.

FInanceCRUD: The financeCRUD class provides operations for managing financial records, including registering, viewing, updating, and removing financial data. It is primarily used by Accountants and Managers to handle financial transactions consistently.

2.5. System Machine State

Figure 2.4: State Machine Diagram for Property Management

UC0001: Use Case for Administration
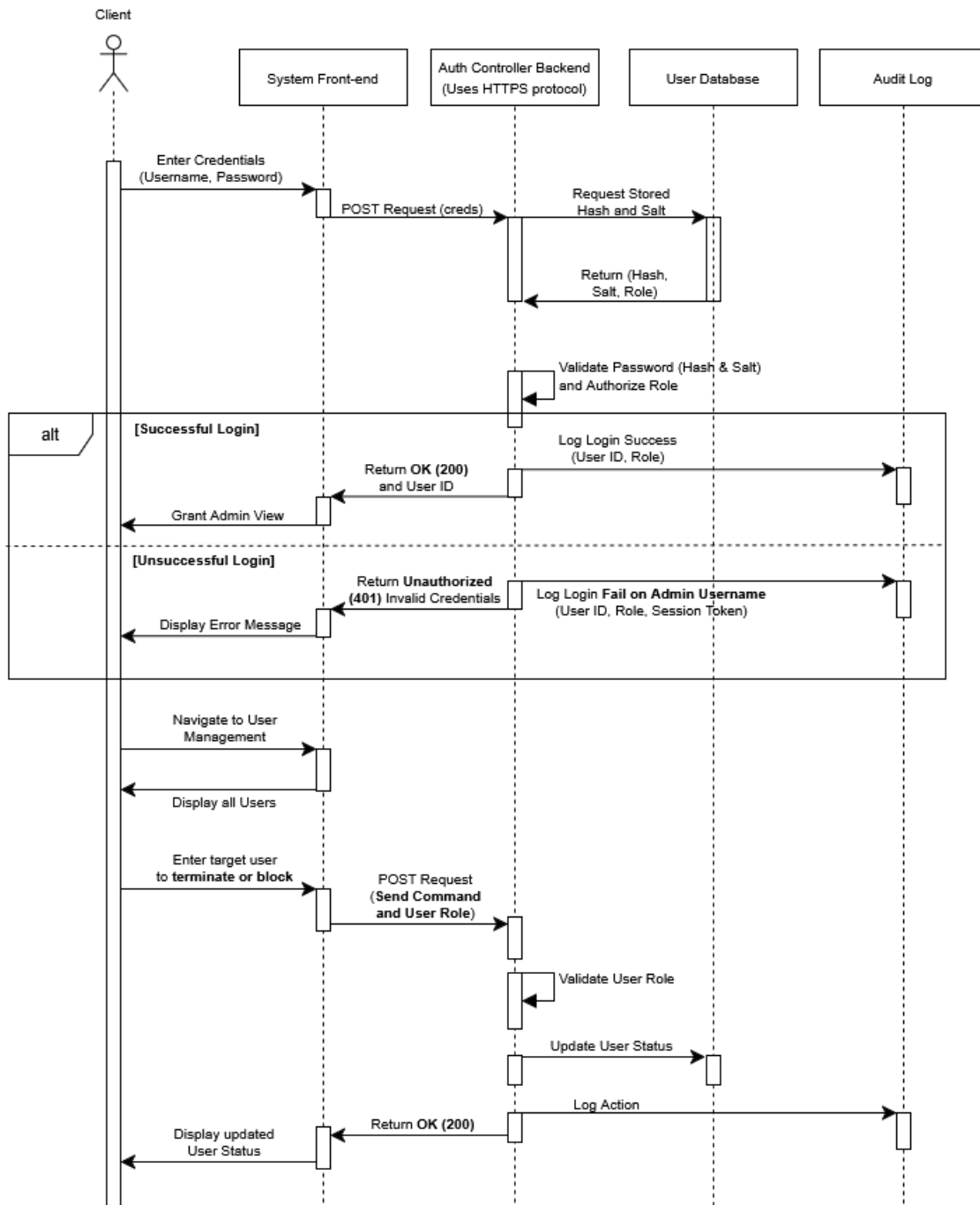
Table 2.1: Use Case Description for UC0001

Use case: System Administration and Access Control
ID: UC0001
Actors: Administrator, Manager
Preconditions: 1. The Actor must have valid credentials (username and password). 2. The Actor's role must be identified as Administrator or Manager.
Flow of events: 1. The Actor accesses the ERP system login page. 2. The Actor enters their valid username and password. 3. The system validates the credentials against the secure data store. 4. The system grants access and checks the Actor's assigned role. 5. The system displays the dashboard/features relevant to the Actor's permissions. 6. (Admin/Manager) The Actor navigates to the User Management section. 7. (Admin) The Actor selects a target user account to terminate or block. 8. The system executes the termination/block command. 9. The system records the action in the audit log.
Postconditions: The Actor is securely logged into the system. Session integrity is maintained. The system has enforced the appropriate Role-Based Access Control (RBAC). If an account was modified, the action is logged and enforced.
Alternative flow: 1. The Actor clicks "Forgot Password". 2. The Actor provides their registered email or username. 3. The system sends a reset link or temporary password. 4. The Actor uses the link/password to create a new, strong password. 5. The Actor provides an invalid password to an admin username (logged).
Postconditions: The Actor's credentials are reset, allowing them to log in, and the new password is stored **hashed/salted**.
Exception flow (if any): **E1: Invalid Login Attempt:** If the user enters an invalid username or password, the system displays an error message.

E2: Too Many Failed Attempts: If the Actor makes too many incorrect attempts, the system blocks the device from entering the site for a specified period of time.

E3: Session Timeout: If the Actor remains inactive for a specified duration, the system automatically logs the user out.

Figure 2.2: Sequence Diagram for UC0001

(Simple Login as Admin and action by updating User status)

UC0002: Use Case Property Management
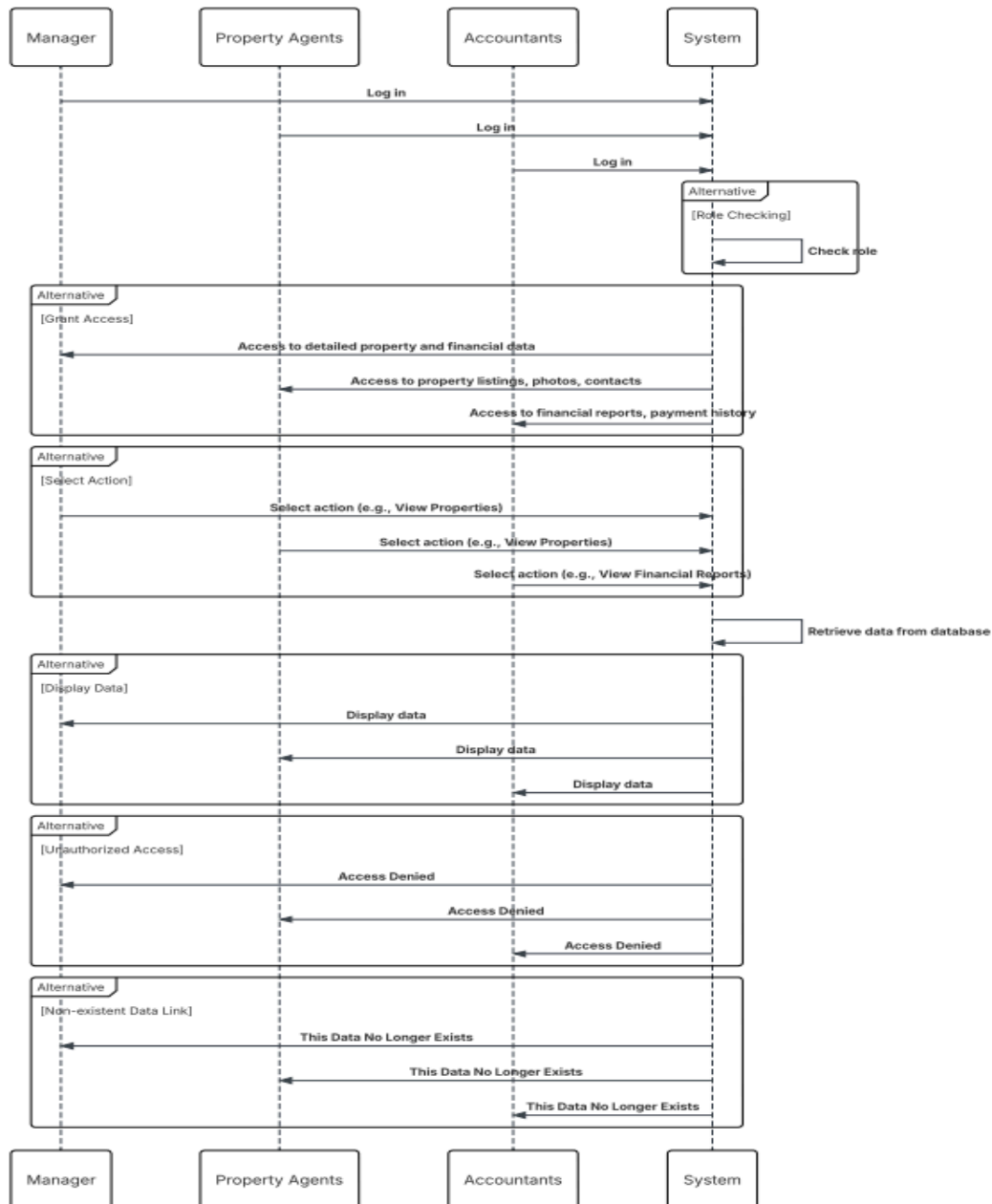
Table 2.3: Use Case Description for UC0002

Use case: Property Management
ID: UC0002
Actors: Manager, Property Agents, Accountants
Preconditions: 1. Role of the actor and that role's permissions must be checked to grant access to the property management system. 2. Property database that have photos and information of properties must be ready and accessible for property agents. 3. Database required for accountants like property values, payments and finance related information must be ready.
Flow of events: 1. The actor opens the system after logging in. 2. System checks the actor's role 3. The system grants access to customized interfaces related to the actor's role. For Property Agents: Grants access to seeing property listings, photos and contact details. For Accountants: Grants access to seeing financial reports, payments history. Does not allow the accountant to view detailed property photos and remarks of property agents. For Manager: Grants access to both detailed property / properties information and financial data. 4. The actor selects an action for example "View Properties" or "View Financial Reports". 5. The System retrieves data from the relevant database. 6. The system displays the requested data in an appropriate format to easily understand.
Postconditions: 1. The actor has successfully viewed the desired information that their role allows them to.
Alternative flow: If the actor has a direct link to a specific data, the system will still allow access after checking the actor's permissions. Following step 6 in flow of events.

Exception flow:
1. Unauthorized Access Attempt: If the actor tries to access data they do not have permission, the system will display "Access Denied".
2. If a link refers to a data that no longer exists, when the actor tries to display it the system will display "This Data No Longer Exists".

Figure 2.4: Sequence Diagram For UC0002
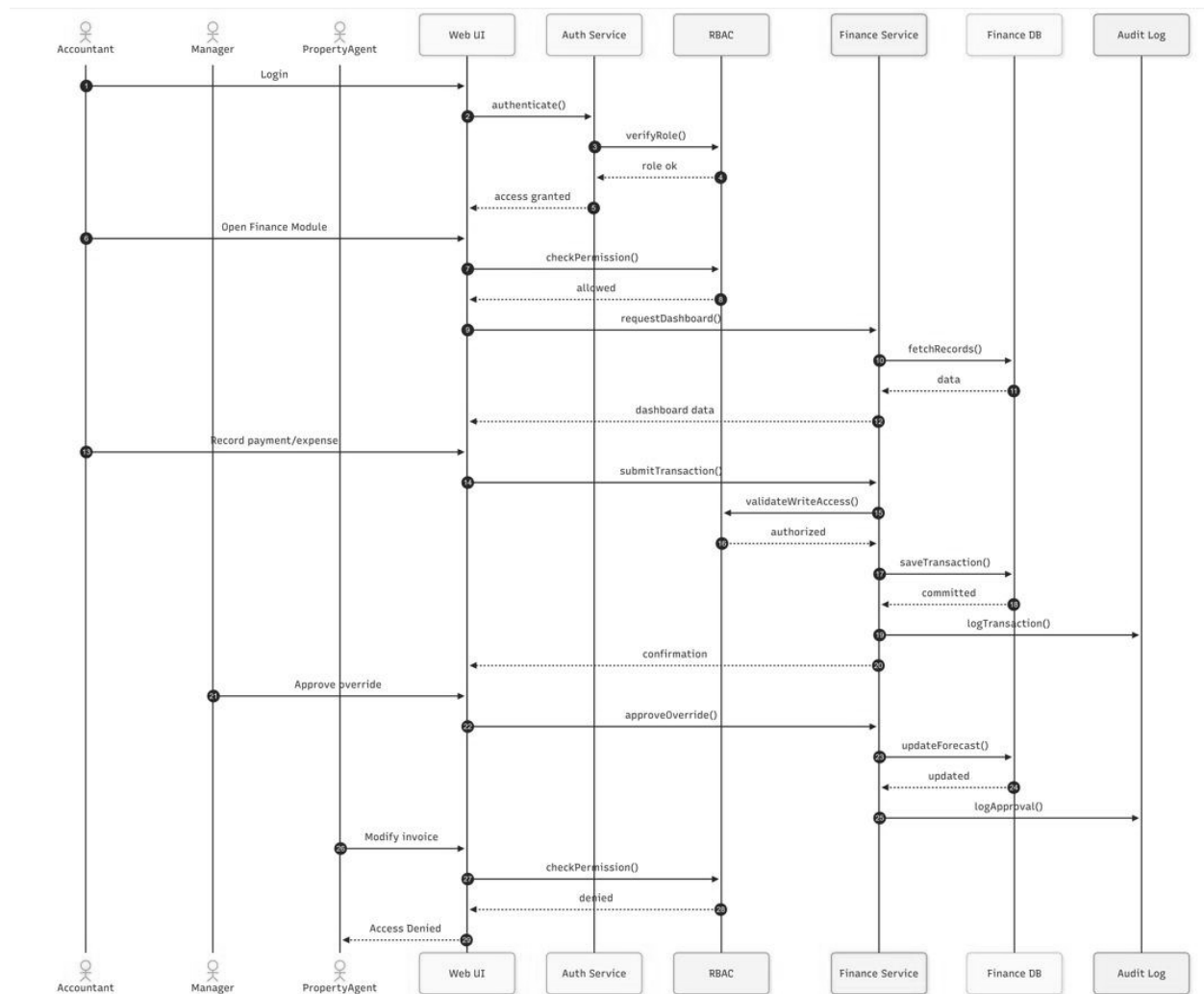
(Viewing data property details and financial data)

UC0003: Use Case for Financial Management

Table 2.4: Use Case Description for UC003

Use case: Financial Management
ID: UC0003
Actors: Accountant, Manager, Property Agent(limited view)
Preconditions: 1. The actor must be logged into the system 2. The actor's role must be verified to grant appropriate access to financial modules. 3. Financial databases (e.g., transactions, invoices, payment schedules) must be populated and accessible. 4. Property data must be linked to financial records for accurate tracking.

Flow of events:
5. The actor logs into the ERP system.
6. The system verifies the actor's role and permissions.
7. Based on role, the system displays the relevant financing dashboard:
    a. **Manager:** Access to financial summaries, budget vs. actuals, approval workflows, and forecasting tools.
    b. **Property Agent:** Read-only access to property-related financials (e.g., commission statements, payment statuses).
8. The actor selects a financial action (e.g., "Record Payment," "Generate Invoice," "View Financial Report").
9. The system retrieves relevant data from the finance database.
10. The system displays the data in a structured format (tables, charts, forms).
11. For write actions (e.g., recording a payment), the system validates inputs, updates records, and logs the transaction.
12. The system confirms the action and updates the dashboard in real time.

Postconditions: 13. Financial records are accurately updated and stored. 14. All transactions are logged for audit purposes. 15. Relevant stakeholders (e.g., managers, accountants) can view updated financial statuses. 16. Role-based access is enforced throughout the session.
Alternative flow: 17. If an actor accesses a direct link to a financial record (e.g., invoice URL), the system will verify permissions before displaying the data.   a. 18. If a manager approves a budget override, the system triggers notifications to accountants and updates forecasting models.
Exception flow: 19. **Unauthorized Access Attempt:** If an actor tries to perform an action outside their permissions (e.g., Property Agent trying to modify an invoice), the system displays "Access Denied."

20. **Data Inconsistency**: If linked property data is missing or invalid during a financial transaction, the system flags the transaction for review and prevents finalization until resolved.
21. **System Timeout:** If the session expires during a financial update, the system rolls back incomplete transactions and requires re-authentication.

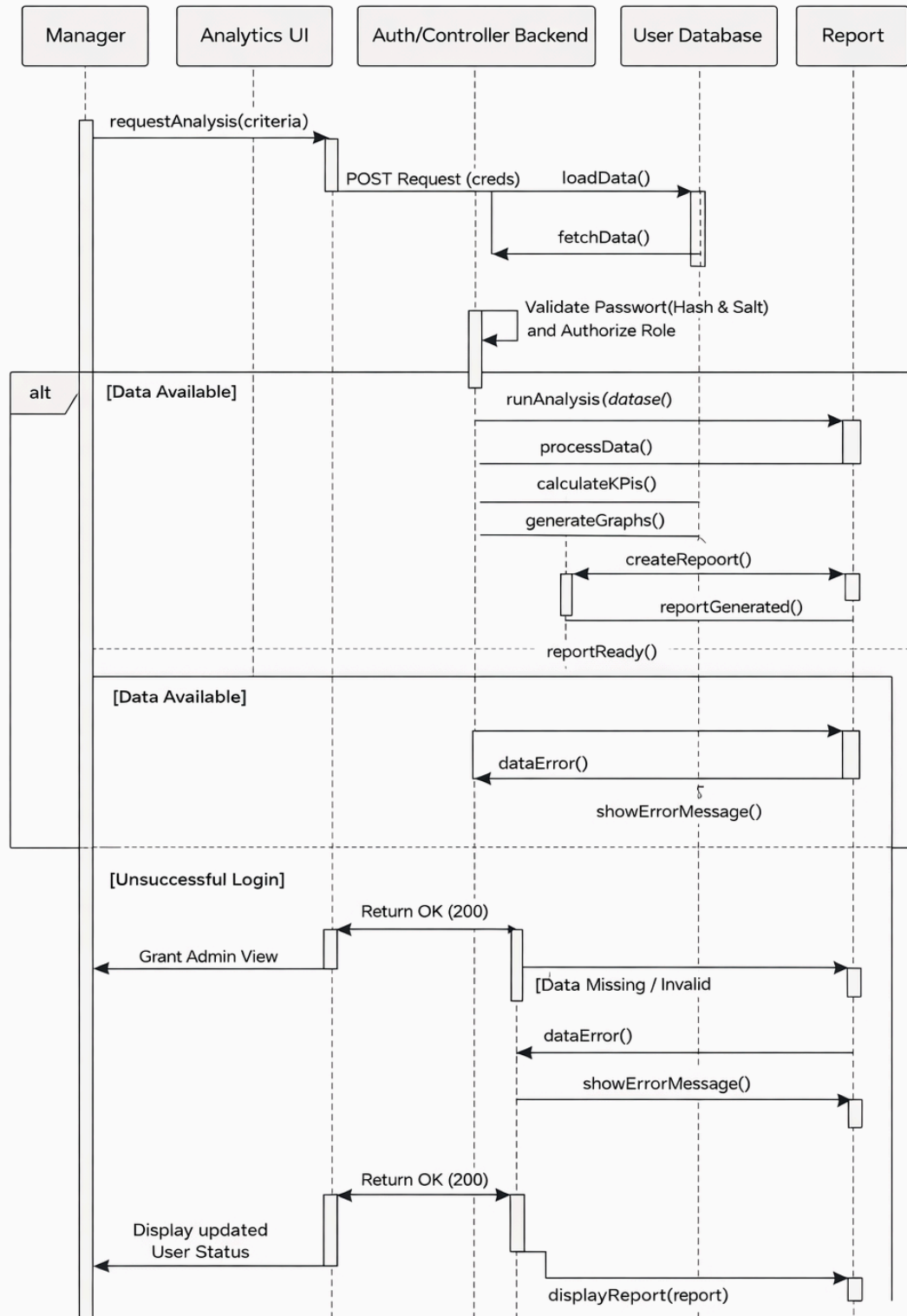**Figure 2.5: Sequence Diagram For UC0003**

UC0004: Use Case Description for Data Analytics

Table 2.5: Use Case Description for UC004

Use case: Data Analytics
ID: UC0004
Actors: Manager, System Administration
Preconditions: Data is first stored in the system. Data has to be cleaned and up to date.
Flow of events: 1. Manager selects the dataset to analyse 2. System then validates and process the data 3. System generates report based on predictive models 4. Manager reviews the report
Postconditions: Operational costs are optimized based on the result of the analysis.
Alternative flow: If the data is inconsistent or missing, the system shall flag the data until the inconsistency is addressed.
Postconditions (alternative flow): The result of the analysis reflects on the accuracy and consistency of the data.
Exception flow: If the inconsistency is ignored despite initial warning, the process goes on and the system will permanently flag the result as one that was built upon missing/inconsistent data.

UC0004: Data Analytics

Manager	Analytics UI	Auth/Controller Backend	User Database	Report

Manager → Analytics UI: requestAnalysis(criteria)

Analytics UI → Auth/Controller Backend: POST Request (creds)

Auth/Controller Backend → User Database: loadData()

User Database → Auth/Controller Backend: fetchData()

Auth/Controller Backend: Validate Passwort(Hash & Salt) and Authorize Role

alt

[Data Available]

Auth/Controller Backend → Report: runAnalysis(*datase*())

Auth/Controller Backend → Report: processData()

calculateKPis()

generateGraphs()

createRepoort()

reportGenerated()

reportReady()

[Data Available]

dataError()

showErrorMessage()

[Unsuccessful Login]

Return OK (200)

Grant Admin View

[Data Missing / Invalid]

dataError()

showErrorMessage()

Return OK (200)

Display updated User Status

displayReport(report)

3. Performance and Other Requirements

This section defines the non-functional requirements (NFRs) of the AM Group ERP system. These requirements specify quality attributes related to system performance, security, usability, reliability, maintainability, and compliance. They complement the functional requirements described in Section 2 by ensuring that the system operates efficiently, securely, and reliably in real-world usage scenarios.

3.1. Performance Requirements

The AM Group ERP system must provide responsive and efficient performance to support multiple concurrent users across different roles.

- **Response Time**
  For standard operations such as login, viewing property records, and retrieving financial reports, the system shall respond within **2 seconds** under normal load conditions.
- **Concurrent Users**
  The system shall support at least **50 concurrent users** without degradation of core functionalities such as property management, financial transactions, and report generation.
- **Data Processing & Analytics**
  Data analytics reports (UC0004) shall be generated within **5 seconds** for standard datasets. Larger datasets shall trigger background processing with progress indicators.
- **Transaction Performance**
  Financial transactions (UC0003) shall be processed atomically, ensuring that data is committed or rolled back entirely to prevent inconsistencies.
- **Scalability**
  The system architecture shall support horizontal scaling to accommodate future growth in users, properties, and financial records.

3.2. Security Requirements

Security is a critical requirement due to the system's handling of sensitive financial and personal data.

- **Authentication & Authorization**
  All users shall be authenticated using unique credentials, and access to system features shall be controlled using Role-Based Access Control (RBAC).
- **Data Encryption**
  All data transmitted between clients and servers shall be encrypted using HTTPS (TLS). Sensitive data such as passwords shall be stored using salted and hashed encryption techniques.
- **Audit Logging**
  The system shall log all critical actions, including login attempts, account modifications, and financial transactions, for audit and compliance purposes.
- **Session Management**
  User sessions shall expire after a predefined period of inactivity, and the system shall prevent session hijacking.

- **Account Lockout**
  The system shall temporarily lock accounts or block devices after repeated failed login attempts.

3.3. Usability Requirements

The ERP system shall be intuitive and accessible to users with varying levels of technical expertise.

- **User Interface Consistency**
  The system shall provide a consistent user interface across all modules and user roles.
- **Accessibility**
  The system shall comply with **WCAG 2.1 AA** accessibility standards, including support for text scaling and high-contrast displays.
- **Role-Specific Views**
  Users shall only see features and information relevant to their assigned roles.
- **Error Feedback**
  Clear and meaningful error messages shall be displayed when user actions fail or input validation errors occur.

3.4. Reliability and Availability Requirements

The system must remain stable and operational during normal business usage.

- **Availability**
  The system shall be available 99% of the time, excluding scheduled maintenance.
- **Fault Tolerance**
  The system shall handle unexpected errors gracefully without data loss.
- **Backup & Recovery**
  Automated daily backups shall be performed, and data restoration shall be possible in the event of system failure.

3.5. Maintainability and Extensibility Requirements

The system shall be designed to allow easy updates and long-term maintenance.

- **Modular Architecture**
  The system shall follow a modular design, separating business logic, data access, and presentation layers.
- **Code Maintainability**
  The system shall follow standard coding conventions and be documented to support future development.
- **System Updates**
  The system shall allow updates and patches to be deployed without affecting existing data.

3.6. Portability and Compatibility Requirements

The ERP system shall operate across different environments without modification.

- **Platform Independence**
  The system shall be accessible through standard web browsers on desktop and mobile devices.
- **Browser Compatibility**
  The system shall support the latest versions of major browsers.

3.7. Compliance and Legal Requirements

The system shall comply with applicable regulations and standards.

- **GDPR Compliance**
  The system shall handle personal data in accordance with GDPR requirements, including data protection and user privacy.
- **Audit Readiness**
  Financial records shall be stored in a manner that supports auditing and traceability.

3.8. Design Constraint

The design and implementation of the AM Group ERP system are constrained by several technical and organizational requirements. The system must be developed as a web-based application to allow users to access it through standard web browsers without the need for local installation.

In addition, the user interface must follow responsive design principles so that the system can be used effectively on desktop computers, tablets, and mobile devices with different screen sizes and resolutions. The system architecture must follow a client–server model that separates presentation, business logic, and data management layers to improve maintainability and scalability. Role-Based Access Control (RBAC) must be enforced across all system modules to ensure that users can only access features appropriate to their roles.

Furthermore, the system must comply with applicable data protection regulations, including the General Data Protection Regulation (GDPR), which imposes constraints on how personal and financial data are stored, processed, and transmitted.

3.9. Software System Testing

Software testing ensures that the AM Group ERP system meets both functional and non-functional requirements and performs reliably in real-world scenarios. Testing activities will be conducted throughout the development lifecycle.

Unit Testing:
Individual modules and components (e.g., authentication, property management, financial transactions, analytics) will be tested in isolation to verify correctness and logic implementation.

System Testing:
The complete system will be tested as an integrated whole to ensure that all modules interact correctly and that end-to-end workflows function as expected.

Performance Testing:
Performance tests will evaluate system responsiveness, scalability, and stability under expected and peak load conditions.

Security Testing:
Security testing will assess authentication, authorization, data encryption, and resistance to common vulnerabilities such as unauthorized access and session attacks.

User Acceptance Testing (UAT):
Selected stakeholders (e.g., administrators, managers, property agents, accountants) will validate that the system meets business needs and usability expectations.

Testing results will be documented, and any identified defects will be addressed before system deployment.

References

[1] Terra Code. (2025). *Project Proposal: PW25 - SE Software Engineering 01 - AM Group GmbH ERP System*. [Terra Code Software Engineering PR 1: Proposal](#).