Software requirements specification (SRS)

Team1

CS-673 (Software Engineering)

Fall 2024

By

Sergio Khalil

Yuhang Zhang

Asma Asiri

Brad Nissenbaum

Shivaang Kumar

Table of Contents:

1. Overview

1.1 Purpose

This project aims to develop a **Real Estate Checklist Tracker** system that streamlines the communication between real estate agents and their clients. This system will help both parties efficiently track the progress of lease agreements, including submitting necessary documents and payments, ensuring transparency, and reducing manual paperwork during the leasing process.

1.2 Benefits

- **Transparency:** Both clients and real estate agents can track the status of leasing applications, reducing confusion and improving communication.
- **Automation:** Automates manual processes such as tracking documents, payments, and statuses, which minimizes errors and miscommunication.
- **Centralized Management**: Provides a single platform for managing all required documents (e.g., proof of income, ID, signed agreements).
- **Efficiency**: Allows agents to monitor the progress of multiple clients simultaneously, while clients can receive real-time updates on their leasing process.
- **Enhanced Decision-Making**: Brokers and agents can make better leasing decisions by efficiently managing and analyzing client data.

1.3 Scope

- **Checklist Creation:** Real estate agents can create and customize checklists based on the leasing needs of individual clients.
- **Real-time Notifications**: Both agents and clients will receive notifications about the progress of tasks, such as completed documents or payments.
- **Document Management**: The system will allow agents and clients to upload, store, and manage documents in a centralized repository.
- **Status Dashboard**: A dashboard will be available to agents and brokers, allowing them to view the progress of their clients' leasing applications.

- **Integration**: The system will integrate with third-party tools for running credit and background checks and uploading results into the platform for further decision-making

2. High-Level Requirements

2.1 Functional Requirements

- **User Authentication and Roles**:
  - The system shall provide secure login and user authentication for brokers, agents, and clients.
  - The system shall assign different roles to users: Brokers, Real Estate Agents, and Clients. Each role will have different access rights.
  - Brokers and agents shall have administrative privileges to create and manage checklists, upload documents, and track the status of multiple clients.
  - Clients shall have access to view and upload documents related to their lease application.

- **Checklist Creation and Management**:
  - The system shall allow real estate agents to create and customize checklists for each client based on the required documents and steps in the leasing process.
  - Agents can modify checklists, add or remove tasks, and assign deadlines.
  - The system shall notify agents and clients in real time whenever a task is updated, completed, or requires attention.

- **Document Upload and Management**:
  - The system shall allow agents and clients to upload documents directly through the platform.
  - All uploaded documents shall be stored in a centralized repository that authorized users can access based on their role.
  - The system shall validate uploaded documents for file type and size, ensuring only approved formats are accepted.

- **Real-Time Notifications**:
  - The system shall send real-time notifications to agents and clients regarding the status of their leasing process, including completing tasks, submitting documents, and pending actions.
  - Notifications shall be sent via email and through the platform's internal messaging system.

- **Dashboard for Monitoring Progress**:
  - Brokers and agents shall have access to a dashboard that provides a high-level overview of the progress of each client's leasing application.
  - The dashboard shall display the status of all tasks, including completed, pending, and overdue items.
  - Brokers shall be able to view the status of multiple agents under their supervision and assess overall business performance.

- **Integration with External Services**:
  - The system shall integrate with third-party services to facilitate credit checks and background verifications, allowing agents to view and manage these reports within the platform.
  - Agents shall be able to upload and store results from these external checks in the document repository.

- **Search and Filtering Capabilities**:
  - The system shall allow agents and brokers to search and filter clients based on various criteria, such as document submission status, checklist completion, and payment status.
  - Clients shall be able to search for their submitted documents and track the progress of their checklist.

- **Data Security and Privacy**:
  - The system shall implement secure data encryption for all stored documents and sensitive information, ensuring compliance with data protection regulations.
  - User access shall be controlled through role-based permissions to ensure only authorized individuals can view and manage documents.

- **Audit and Logging**:
  - The system shall maintain logs of all user interactions, including document uploads, checklist modifications, and status changes.
  - These logs shall be accessible to brokers and agents for review, ensuring accountability and transparency in the leasing process.

2.2 Non-Functional Requirements

- **Performance Requirements:**
  - **Response Time**: The system shall respond to user actions, such as document uploads, checklist updates, or dashboard loading, within 2 seconds under normal operating conditions. The system shall handle searches and data retrieval (e.g., filtering client information) with a maximum response time of 3 seconds.
  - **Scalability**: The system shall support up to 1000 concurrent users without significant degradation in performance. The architecture shall scale vertically and horizontally to accommodate increasing users and data as the client base grows.

- **Security Requirements:**
  - **Data Encryption**: All sensitive data, including user credentials, documents, and personal information, shall be encrypted using AES-256 encryption or higher.
  - **User Authentication**: The system shall implement secure authentication mechanisms, including two-factor authentication (2FA)for agents and brokers to ensure authorized access to the system.
  - **Role-Based Access Control**: The system shall enforce role-based access control, ensuring that users (e.g., agents, brokers, and clients) can only access data and functionalities pertinent to their roles.
  - **Compliance**: The system shall comply with relevant data protection and privacy regulations, such as GDPR and HIPAA(for U.S.-based users), ensuring that all stored and processed personal data is handled securely and transparently.

- **Usability Requirements:**
  - **User Interface**: The user interface shall be intuitive and easy for tech-savvy and non-technical users, with a clear distinction between tasks, statuses, and notifications.
  - **Multi-Device Support**: The system shall be fully responsive, allowing access from various devices, such as desktops, tablets, and mobile phones, with consistent performance across platforms.
  - **Language Support**: The system shall support multiple languages, allowing users to switch between supported languages in the user interface to enhance accessibility.

- **Reliability and Availability:**
  - **Uptime**: The system shall be available 99.9% of the time, ensuring minimal user downtime.
  - **Data Backup**: Daily backups of the system's database and user data shall be conducted to prevent data loss, and backups shall be stored securely to restore the system within 2 hours in case of failure.

- **Maintainability and Support:**
  - **Code Maintainability**: The system's codebase shall follow standardized coding conventions, and all new code shall undergo code reviews to ensure it is clean, well-documented, and easy to maintain.
  - **Version Control**: A Git-based version control system (e.g., GitHub) shall be used to manage the source code, and every release shall include well-documented features, bug fixes, and updates.

- **Compatibility Requirements:**

The system shall be compatible with all major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, and should maintain full functionality across all browsers.

- **Disaster Recovery:**

In case of a critical system failure, the system shall be recoverable within 4 hours, with minimal data loss through regular backups and redundancies.

- **Logging and Monitoring:**

The system shall keep audit logs of all users' significant actions (e.g., document uploads and task completions) for at least 12 months to provide traceability and accountability. Continuous monitoring tools shall be implemented to track system performance, security breaches, and resource usage, with alerts sent to administrators in case of anomalies.

- **Scalability Requirements:**

The system shall be designed to scale horizontally by adding more servers and vertically by enhancing the system's hardware to accommodate growing numbers of users and higher traffic.

2.3 Use cases or user stories for all requirements

1. Use Case Name: User Authentication and Role Assignment

Actors: Brokers, Agents, Clients.
Precondition: The user has registered in the system and has valid credentials.

Basic Flow:

1. The user navigates to the login page.
2. The user enters their username and password.
3. The system verifies the credentials and authenticates the user.
4. The system identifies the user's role (broker, agent, client).
5. The system redirects the user to the appropriate dashboard based on their role.

Alternative Flows:

- **Invalid Credentials**:
    - If the username or password is incorrect, the system displays an error message and prompts the user to retry.
- **Account Locked**:
    - If the user fails to authenticate after three attempts, the system locks the account for a defined period and displays a message to contact support.

2. Use Case Name: Create a Checklist for a Client

Actors: Real Estate Agent.
Precondition: The agent is logged into the system with valid credentials.

Basic Flow:

1. The agent navigates to the "Create New Checklist" section.
2. The agent enters the client's details.
3. The agent selects the document and task requirements for the client (e.g., proof of income, photo ID).
4. The agent sets deadlines for each task.
5. The agent saves and publishes the checklist.
6. The system notifies the client of the new checklist and requirements.

Alternative Flows:

- **Incomplete Information**:
    - The system displays a warning if the agent fails to enter all necessary details (e.g., missing client information). It prevents the checklist from being saved until the required fields are completed.
- **Checklist Cancelled**:
    - The agent can choose to cancel the checklist creation process at any time. The system discards the entered data and returns the agent to the dashboard.

3. Use Case Name: Uploading Documents

Actors: Clients, Agents.
Precondition: The checklist has been created, and the client has been notified of the required documents.

Basic Flow:

1. The client logs into the system and accesses the checklist.
2. The client selects a task requiring document submission.
3. The client uploads a document by selecting a file from their computer or dragging it into the system.
4. The system verifies the file type and size.
5. The document is uploaded and linked to the corresponding task in the checklist.
6. The agent is notified of the new document upload.

7. The agent reviews the document and marks the checklist item as completed.

Alternative Flows:

- **Invalid File Format or Size**:
    - If the uploaded file is of an invalid format or exceeds the size limit, the system rejects the upload and prompts the client to upload a valid file.
- **Upload Failure**:
    - If the document upload fails due to connectivity issues, the system informs the client and prompts them to try again later.

4. Use Case Name: Monitor the Progress of Agents' Clients

Actors: Broker.
Precondition: The broker is logged into the system and can access the agents' data.

Basic Flow:

1. The broker navigates to the "Dashboard" section.
2. The system displays an overview of all agents and their clients.
3. The broker views the progress of each client (e.g., document submission, task status).
4. The broker filters clients by status (e.g., pending, completed).
5. The broker views detailed reports on agent performance and client progress.

Alternative Flows:

- **No Clients Assigned**:
    - If an agent has no active clients, the system displays a message indicating no data is available for that agent.
- **Delayed Status Updates**:
    - If real-time updates are delayed due to system issues, the broker is notified and given an option to refresh the data manually.

5. Use Case Name: Integration with Third-Party Services for Credit Checks

Actors: Real Estate Agent.
Precondition: The agent is logged in and can access the client's checklist.

Basic Flow:

1. The agent selects the client from the checklist view.
2. The agent clicks on the "Run Credit Check" button.
3. The system sends a request to the third-party credit check service.
4. The credit check is processed, and the results are automatically uploaded to the client's checklist.
5. The agent reviews the credit check results.

Alternative Flows:

- **Service Unavailable**:
    - If the third-party credit check service is unavailable, the system informs the agent and allows them to retry later.
- **Credit Check Failure**:
    - If the credit check fails due to incorrect data or service issues, the system displays an error message and requests the agent to verify the information.

6. Use Case Name: Track Leasing Application Progress in Real-Time

Actors: Client.
Precondition: The client is logged into the system, and a checklist has been created for their leasing application.

Basic Flow:

1. The client logs into the system and accesses the checklist.
2. The client views the checklist items and their status (e.g., pending, completed).
3. The system displays a progress bar indicating the overall progress of the application.
4. The client receives notifications about upcoming deadlines and task completions.
5. The client can download previously uploaded documents if needed.

Alternative Flows:

- **No Tasks Assigned**:
    - If no tasks have been assigned, the system informs the client that the checklist is yet to be available.
- **Missed Deadline**:
    - If a task deadline is missed, the system sends a notification to the client and the agent, indicating that the task is overdue.

7. Use Case Name: System Notifications for Task Completion

Actors: Agents, Clients.
Precondition: The checklist has been created, and tasks are assigned with deadlines.

Basic Flow:

1. The client uploads a document or completes a task.
2. The system notifies the agent of the completed task.
3. The agent reviews the task and marks it as complete.
4. The system sends a confirmation notification to the client.
5. The system sends reminders to the client and agent when task deadlines are approaching.

Alternative Flows:

- **Notification Failure**:
    - If the notification system fails, users are prompted to check their task statuses manually by logging into the system.
- **Deadline Extension**:
    - If the agent extends the task deadline, the system updates the client with the new deadline and sends a notification.

8. Use Case Name: Search for a Client and View Their Progress

Actors: Real Estate Agent.
Precondition: The agent is logged into the system and can access client data.

Basic Flow:

1. The agent uses the search bar to enter the client's name or ID.
2. The system displays the matching client profiles.
3. The agent selects the client from the search results.
4. The agent views the client's checklist and task progress.
5. The agent filters the checklist items by status (e.g., pending, completed).

Alternative Flows:

- **No Matching Client**:
    - If no matching client is found, the system displays a message indicating no results.
- **Client Data Unavailable**:
    - If the client's data is temporarily unavailable due to system issues, the system informs the agent and prompts them to try again later.

9. Use Case Name: Ensure System Reliability and Security

Actors: System Administrator.
Precondition: The system is operational, and regular monitoring is in place.

Basic Flow:

1. The system administrator monitors the system's performance metrics.
2. The system logs critical events (e.g., document uploads and task completions).
3. The system runs daily data backups.
4. The system alerts the administrator of any anomalies or performance issues.
5. The system automatically recovers from failures within 4 hours, ensuring data integrity.

Alternative Flows:

- **Backup Failure**:
    - If a data backup fails, the system retries the backup and informs the system administrator of the issue.

- **Security Breach**:
    - If the system detects a security breach, the system alerts the administrator and locks down the affected accounts.

3. High-level interface (GUI) designs

The Real Estate Checklist Tracker GUI is designed to provide a user-friendly and efficient interface for clients, agents, and brokers. The GUI will ensure intuitive navigation, real-time updates, and role-based functionality, offering a seamless experience for managing leasing applications, tasks, and documents.

3.1 Graphical User Interface (GUI)

The GUI will cater to the different user roles, clients, agents, and brokers, each with distinct views and actions specific to their responsibilities. The design emphasizes clarity, accessibility, and ease of use.

3.1.1 Login Screen

- **Purpose**: Securely authenticate users and direct them to their respective dashboards based on their role (client, agent, broker).
- **Elements**:
    - **Username and Password Fields**: Input fields for user credentials.
    - **Login Button**: Authenticates the user and grants access to the system.
    - **Forgot Password Link**: Allows users to reset their password if needed.
    - **Error Notification**: Displays error messages for incorrect login attempts.

3.1.2 Dashboard (For Agents and Brokers)

- **Purpose**: Centralized management view for agents and brokers to monitor client progress, manage checklists, and track document submissions.
- **Elements**:
    - **Task List**: Displays tasks for each client, categorized by status (e.g., Pending, In Progress, Completed).
    - **Client Overview**: A summary of each client's checklist progress, including submitted documents and completed tasks.
    - **Search Bar**: Allows agents or brokers to search for specific clients by name or ID.
    - **Filter Options**: Enables filtering of tasks or clients based on status (e.g., pending tasks, overdue tasks).
    - **Notification Panel**: Real-time alerts for task updates, new document uploads, and approaching deadlines.
    - **Navigation Menu**: Provides access to different platform sections, such as Checklists, Documents, Reports, and Settings.

3.1.3 Client Interface

- **Purpose**: Allows clients to track the progress of their leasing application, view and complete checklist tasks, as well as upload required documents.
- **Elements**:
    - **Checklist View**: Displays tasks with status indicators (e.g., green for completed, red for overdue), showing clients what documents or actions are required.
    - **Document Upload Section**: Clients can upload documents by dragging and dropping files or selecting them through a file browser.
    - **Progress Bar**: A visual indicator of the client's overall progress in completing the checklist.
    - **Notifications**: Displays system notifications for upcoming deadlines, submitted documents, and task updates.
    - **Download Section**: Clients can download previously uploaded documents or view shared documents from agents.

3.1.4 Document Management Interface

- **Purpose**: Allows clients and agents to manage and review documents related to leasing applications.
- **Elements**:
    - **Upload Area**: A drag-and-drop or file selection interface for uploading documents.
    - **File Preview**: Allows users to preview the document before submitting it.
    - **Status Labels**: Shows the document status (e.g., "Uploaded," "Pending Review," "Approved").
    - **Upload History**: Displays a list of previously uploaded documents with timestamps for tracking purposes.

3.1.5 Admin Interface (For Brokers)

- **Purpose**: Provides brokers with an administrative view to monitor agent performance and manage the system.
- **Elements**:
    - **Agent Summary**: A dashboard showing agent performance metrics (e.g., number of clients managed, tasks completed, tasks overdue).
    - **Reports Section**: Tools for generating performance reports based on criteria such as client progress or task completion rates.
    - **User Management**: Allows brokers to manage user roles, add or remove agents, and set permissions.
    - **System Logs**: A log of system events such as document uploads, task completions, and user logins, allowing for auditing and accountability.

3.1.6 Notification System

- **Purpose**: Keep users (clients, agents, brokers) informed of important events such as document uploads, task completions, and upcoming deadlines.
- **Elements**:
    - **In-App Alerts**: Real-time pop-up notifications for critical events (e.g., "Document uploaded by Client X").
    - **Notification Center**: A section where users can review recent notifications in case they missed any alerts during their session.
    - **Settings**: Allows users to customize their notification preferences (e.g., frequency of notifications, types of events).

3.2 Navigation and Interaction Design

- **Primary navigation:** A sidebar or top navigation bar will provide access to the main sections of the system, including the Dashboard, Documents, reports, and settings. The navigation will be tailored based on the user roles, showing clients, agents, and brokers different options.

- **Interactive elements:** Button for actions such as creating a checklist, uploading documents, and marking as completed will be present throughout the system, providing clear, responsive interaction points. Dropdown menus and filters allow users to refine task views and search for specific clients or documents.

- **Responsive design:** The interface will adapt to different screen sizes, ensuring usability across devices such as desktops, tablets, and smartphones. This flexibility ensures users can access the system and perform tasks from any device.