**High-Level Design (HLD):**

* **User Authentication and Authorization**

1. **Overview:**
   * The User Authentication and Authorization module is responsible for verifying the identity of users and granting appropriate access permissions based on their roles.
2. **Components:**
   * **Authentication Server:** Handles user authentication by verifying credentials (username/email and password) against stored user data.
   * **Authorization Server:** Manages access control by enforcing permissions based on user roles and privileges.
   * **User Database:** Stores user credentials, roles, and permissions securely.
3. **Architecture:**
   * **Client-Server Architecture:** Clients (user interfaces) interact with Authentication and Authorization servers to authenticate users and authorize access to resources.
   * **Token-Based Authentication:** After successful authentication, the server issues a token (e.g., JWT) to the client, which is used to authenticate subsequent requests.
4. **Workflow:**
   * **User Authentication:**
     + User submits credentials via the login form.
     + Authentication server verifies credentials against the user database.
     + Upon successful authentication, the server issues an authentication token to the client.
   * **User Authorization:**
     + Client includes the authentication token in subsequent requests.
     + Authorization server validates the token and checks user roles/permissions.
     + Authorized requests are processed, and access is granted to requested resources.

**User Stories**

1. **As a user, I want to be able to register for an account, so that I can access the accounting software securely.**
   * Accept user registration details such as email address and password.
   * Validate user input and create a new account in the system.
2. **As a user, I want to be able to log in to my account, so that I can access the accounting software and my data.**
   * Provide a login form where users can enter their credentials (email and password).
   * Verify user credentials and authenticate the user.
   * Redirect authenticated users to the dashboard or landing page.
3. **As a user, I want the option to reset my password if I forget it, so that I can regain access to my account.**
   * Provide a "Forgot Password" link on the login page.
   * Allow users to enter their email address to receive a password reset link.
   * Send an email with a unique password reset link to the user's email address.
   * Implement a secure password reset mechanism to update the user's password.
4. **As a user, I want to see personalized content and features based on my role and permissions, so that I only have access to the features relevant to me.**
   * Customize the user interface based on the user's role and permissions.
   * Hide or disable features that the user does not have permission to access.
5. **As a user, I want to be able to securely log out of my account, so that I can protect my data and privacy.**
   * Provide a "Log Out" option in the user interface.
   * Redirect the user to the login page after successfully logging out.