Book Alley

Use-Case Specification: Manage Users

Version 2.0

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 14/11/2023 | 1.0 | Initial | Dang Ha Huy |
| 17/11/2023 | 2.0 | Review | Chau Tan Kiet |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Use-Case Name 4**](#_heading=h.30j0zll)

[1.1 Brief Description 4](#_heading=h.1fob9te)

[**2. Flow of Events 4**](#_heading=h.3znysh7)

[2.1 Basic Flow 4](#_heading=h.2et92p0)

[2.2 Alternative Flows 4](#_heading=)

[2.2.1 User Creation Errors 4](#_heading=h.3dy6vkm)

[2.2.2 Password Reset Challenges 4](#_heading=h.dtzb0ty3x9qs)

[**3. Special Requirements 4**](#_heading=h.4d34og8)

[3.1 Password Policy Enforcement 4](#_heading=h.2s8eyo1)

[3.2 User Data Privacy 4](#_heading=h.w5h01a98mda8)

[3.3 Emergency Access Procedures 4](#_heading=h.6kfk4vz0ogox)

[**4. Preconditions 4**](#_heading=h.17dp8vu)

[4.1 Authentication 4](#_heading=h.3rdcrjn)

[4.2 Administrator Privileges 4](#_heading=h.4mp9ou5heqce)

[**5. Postconditions 5**](#_heading=h.26in1rg)

[5.1 User Account Changes Persist 5](#_heading=h.lnxbz9)

[5.2 User Role Assignment 5](#_heading=h.tas3w6wyk8y)

[**6. Extension Points 5**](#_heading=h.35nkun2)

[6.1 User Authentication Extension Point 5](#_heading=h.1ksv4uv)

[6.2 Custom User Fields 5](#_heading=h.u99oura6v21f)

Use-Case Specification: Manage Users

# Use-Case Name

## Brief Description

The "Manage Users" use case for an administrator involves the efficient and effective handling of user-related activities within a system or platform. Administrators are responsible for overseeing and controlling user accounts to ensure secure and organized access.

# Flow of Events

## Basic Flow

1. The administrator selects the option to create a new user.
2. The administrator assigns a role and sets appropriate permissions.
3. The administrator selects a user requesting a password reset.
4. The administrator chooses to suspend or reactivate the account based on the need.

## Alternative Flows

### User Creation Errors

If there are errors in the user creation process (e.g., duplicate username or invalid email), the system prompts the administrator to correct the issues before saving.

### Password Reset Challenges

If the user faces difficulties with the password reset process, the admin may choose alternative methods (manual reset, contacting support) to ensure access is regained securely.

# Special Requirements

## Password Policy Enforcement

* Complexity Requirements: Enforce password complexity requirements to ensure that passwords are strong and resistant to brute force attacks. These requirements may include minimum length, mandatory inclusion of different character types (uppercase letters, lowercase letters, numbers, symbols), and restrictions on consecutive identical characters.
* Regular Password Updates: Mandate regular password updates to prevent prolonged use of compromised passwords. Implement a grace period for password changes to minimize user inconvenience while maintaining security.
* Password Reuse Prevention: Prevent password reuse to discourage users from using the same password across multiple accounts. Implement a mechanism to check for password reuse and prompt users to change their passwords if necessary.

## User Data Privacy

* Data Encryption: Encrypt sensitive user data, such as personal information, payment details, and browsing history, at rest and in transit. Utilize industry-standard encryption algorithms to protect data confidentiality.
* Data Collection and Use: Clearly inform users about the types of data collected, the purposes for data collection, and how the data will be used. Obtain explicit consent from users before collecting and using their data.
* Data Minimization: Collect only the minimum amount of data necessary for the intended purpose. Avoid collecting unnecessary or irrelevant data that may increase privacy risks.

## Emergency Access Procedures

* Authorization and Approval Process: Establish a structured authorization and approval process for granting emergency access to critical user accounts. Require multiple levels of approval from authorized personnel to ensure proper oversight and accountability.
* Logging and Auditing: Implement comprehensive logging and auditing mechanisms to track emergency access events. Maintain detailed records of who accessed the account, when access was granted, and the actions performed during the access period.
* Time-Bound Access: Limit emergency access to a specific timeframe or until the emergency situation is resolved. Automatically revoke emergency access privileges after the designated period or upon resolution of the emergency.

# Preconditions

## Authentication

* Ensuring proper authentication is crucial for safeguarding user management functions and preventing unauthorized access to sensitive user data. This precondition emphasizes the importance of implementing a secure login process that verifies the identity of the administrator attempting to access user management functionalities.

1. Administrators must log in securely before accessing user management functions.
2. Use strong authentication protocols, such as multi-factor authentication (MFA).

## Administrator Privileges

* The user account used for the administrator login must have the necessary privileges and permissions to access and modify user management settings.

1. Only authorized administrators with necessary permissions can access user management settings.
2. Implement role-based access control to manage user privileges effectively.
3. Adhere to the principle of least privilege, granting administrators only the minimum level of access required

# Postconditions

## User Account Changes Persist

* When an administrator creates a new account, updates a profile, or changes permissions, those changes should be reflected in the system and remain in effect until they are explicitly reversed.
* To ensure that user account changes persist, the system should implement a robust data storage mechanism that can reliably store and retrieve user data. The system should also have mechanisms in place to prevent data loss or corruption. For example, the system could use regular backups and data integrity checks to protect against data loss.

## User Role Assignment

* A user management system should be able to assign or modify roles for specified users. Roles are used to define the permissions that a user has within the system. For example, a user with an administrator role might have the permission to create new accounts, while a user with a standard user role might only have the permission to view their own account information.
* To assign or modify roles for users, the system should have a role management interface that allows administrators to select the appropriate roles for each user. The system should also have mechanisms in place to prevent unauthorized role changes. For example, the system could require administrators to enter a password or use two-factor authentication to change user roles.

# Extension Points

## User Authentication Extension Point

* A user authentication extension point allows the integration of external authentication providers like OAuth and LDAP into the user management system. This enables the system to support a wider range of authentication mechanisms, catering to diverse user preferences and security requirements.
* To implement a user authentication extension point, the system should provide a mechanism for registering and configuring external authentication providers. This could involve defining a configuration interface or using a plugin-based architecture. Once an external authentication provider is registered, the system should be able to seamlessly delegate authentication requests to the appropriate provider.

## Custom User Fields

* A custom user fields extension point allows the addition of custom fields to user profiles without modifying the core user management functionality. This provides the flexibility to accommodate specific requirements of different applications or organizations.
* To implement a custom user fields extension point, the system should provide a mechanism for defining and managing custom fields. This could involve a configuration interface or a plugin-based architecture. Once a custom field is defined, the system should be able to store and retrieve user data associated with that field.