Text Description:-

**UC 1: Login.** Using this use case, the user/administrator logs into the software for using it.

*Actors*: User, Administrator

**Stakeholders and interests**

*User*: Wants a user-friendly interface and proper steps for maintaining the office stockbook.

*Administrator*: Wants to keep a track of all the transactions carried out in the stockbook. Also can maintain the office stockbook.

*Pre-condition*: The homepage is displayed with Login form.

*Post-conditions*: The user/administrator successfully logs into the software.

*Primary actor:* User

**Main success scenario**

1. System displays a login form.
2. User selects their respective domain.
3. User enters username and password and clicks “Login” button.
4. System validates the username and password and logs into the software.
5. System enters the information about user, domain and login information in the transaction log.
6. System checks the last software database scheduled backup date and creates a backup if the last scheduled backup date is seven days older.
7. System enters the scheduled backup information in the transaction log.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid username or password: | Handle invalid username or password error. |
| 3.b. | Empty field: | Handle empty field error. |
| 3.c. | Incomplete form submission: | Handle incomplete form submission. |
| 4.a. | Invalid login details: | Handle invalid login details. |
|  |  |  |

*Primary actor:* User

**Alternate success scenario 1**

1. System displays a login form.
2. User selects their respective domain.
3. User enters username and password.
4. User checks the ‘Change login details’ option and clicks “Login” button.
5. System validates the username and password and logs into the software.
6. System enters the information about user, domain and login information in the transaction log.
7. System checks the last software database scheduled backup date and creates a backup if the last scheduled backup date is seven days older.
8. System enters the scheduled backup information in the transaction log.
9. System displays a form for changing the login password.
10. User enters the required details and clicks “Save” button.
11. System validates the old password and saves the new password.
12. System enters the information about user, domain and password change information in the transaction log.
13. System displays ‘Password Change successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid username or password: | Handle invalid username or password error. |
| 3.b. | Empty field: | Handle empty field error. |
| 4.a. | Incomplete form submission: | Handle incomplete form. |
| 5.a. | Invalid login details: | Handle invalid login details. |
| 10.a. | Incomplete form submission: | Handle incomplete form. |
| 11.a. | Invalid old password: | Handle invalid old password. |
| 11.b. | Failure to save password: | Handle failure to save new password. |

*Primary actor:* Administrator

**Alternate success scenario 2**

1. System displays a login form.
2. Administrator selects the administrator domain.
3. Administrator enters username and password and clicks “Login” button.
4. System validates the username and password and logs into the software.
5. System enters the information about user, domain and login information in the transaction log.
6. System checks the last software database scheduled backup date and creates a backup if the last scheduled backup date is seven days older.
7. System enters the scheduled backup information in the transaction log.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid username or password: | Handle invalid username or password error. |
| 3.b. | Empty field: | Handle empty field error. |
| 3.c. | Incomplete form submission: | Handle incomplete form submission. |
| 4.a. | Invalid login details: | Handle invalid login details. |

*Primary actor:* Administrator

**Alternate success scenario 3**

1. System displays a login form.
2. Administrator selects the administrator domain.
3. Administrator enters username and password.
4. Administrator checks the ‘Change login details’ option and clicks “Login” button.
5. System validates the username and password and logs into the software.
6. System enters the information about user, domain and login information in the transaction log.
7. System checks the last software database scheduled backup date and creates a backup if the last scheduled backup date is seven days older.
8. System enters the scheduled backup information in the transaction log.
9. System displays a form for changing the login password.
10. Administrator enters the required details and clicks “Save” button.
11. System validates the old password and saves the new password.
12. System enters the information about user, domain and password change information in the transaction log.
13. System displays ‘Password Change successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid username or password: | Handle invalid username or password error. |
| 3.b. | Empty field: | Handle empty field error. |
| 4.a. | Incomplete form submission: | Handle incomplete form. |
| 5.a. | Invalid login details: | Handle invalid login details. |
| 10.a. | Incomplete form submission: | Handle incomplete form. |
| 11.a. | Invalid old password: | Handle invalid old password. |
| 11.b. | Failure to save password: | Handle failure to save new password. |

**UC 2: KVK Stockbook.** Using this use case the user/administrator can manage critical inputs, manage the stockbook records and also take stockbook statements for a selected time period.

**UC 2.1: Critical Input.** Using this use case the user/administrator can manage the critical inputs of the stockbook.

**UC 2.1.1: New Critical Input.** Using this use case the user/administrator can create new critical input in stockbook.

*Actors:* User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface for creating new critical input in stockbook.

*Pre-condition*: The homepage is displayed with ‘New Critical Input’ option under ‘Critical Input’ under ‘KVK Stockbook’, for creating a new critical input.

*Post-conditions*: The user/administrator successfully creates a new critical input.

**Main success scenario**

1. User/Administrator clicks the ‘New Critical Input’ option in the homepage “START” menu bar.
2. System opens a form for creating new critical input.
3. User/Administrator enters critical input details and clicks “Create” button.
4. System saves the new critical input details in database.
5. System enters the information about user, domain and new critical input create information in the transaction log.
6. System displays “New critical input created successfully” message.
7. User/Administrator clicks “Cancel” button.
8. System closes the form for creating new critical input.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid critical input details: | Handle invalid details error. |
| 3.b. | Critical input already exist: | Handle already exist error. |
| 3.c. | Empty field: | Handle empty field error. |
| 3.d. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | Failure to save details: | Handle failure to save. |

**Special requirement**

The User/Administrator must be careful while entering the critical input details as, once created, it cannot be edited. A critical input must be deleted and a new critical input must be created in case of any mistake. If any stockbook transaction is associated with the critical input then it cannot be deleted. The transactions are to be deleted before deleting the critical input. All these types of operations are logged in the transaction log and the user will be held answerable for these operation.

**UC 2.1.2: Delete Critical Input.** Using this use case the administrator can delete a critical input in stockbook.

*Actors:* User, Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface creating new critical input in stockbook.

*Pre-condition*: The homepage is displayed with ‘Delete Critical Input’ option under ‘Critical Input’ under ‘KVK Stockbook’, for deleting a critical input.

*Post-conditions*: The administrator successfully creates a new critical input.

*Primary actor*: Administrator

**Main success scenario**

1. Administrator clicks the ‘Delete Critical Input’ option in the homepage “START” menu bar.
2. System authenticates the administrator and opens the form for deleting critical input.
3. Administrator selects the critical input to be deleted form the dropdown list and clicks “Delete” button.
4. System checks whether any stockbook transaction is associated with the selected critical input.
5. System deletes the critical input details from the database.
6. System enters the information about user, domain and critical input delete information in the transaction log.
7. System displays “Critical input deleted successfully” message.
8. Administrator clicks “Cancel” button.
9. System closes the form for deleting critical input.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 2.a. | Invalid authentication | Handle error, prevent delete form open. |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | Critical input transaction exists: | Handle error, prevent delete. |
| 3.c. | Empty field: | Handle empty field error. |
| 5.a. | Failure to delete critical input: | Handle failure to save. |

**Special requirement**

The Administrator must be careful while deleting the critical input details. If any stockbook transaction is associated with the critical input then it cannot be deleted. The transactions are to be deleted before deleting the critical input. All these types of operations are logged in the transaction log and the user will be held answerable for these operation.

*Primary actor*: User

**Alternate success scenario**

1. User clicks the ‘Delete Critical Input’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to delete a critical input.

**UC 2.2: Stock Details.** Using this use case the user/administrator can manage the stockbook details like Update, Edit and Delete stockbook records.

**UC 2.2.1: Stock Detail Update.** Using this use case the user/administrator can create a new stockbook entry for a day.

*Primary actors:* User, Administrator

**Stakeholders and interests**

*User*/*Administrator*: Wants a user-friendly interface for updating a new stockbook entry for the day.

*Pre-condition*: The homepage is displayed with ‘Stock Detail Update’ option under ‘Stock Details’ under ‘KVK Stockbook’, for creating a new stockbook entry.

*Post-conditions*: The stockbook entry is created with proper stockbook details.

**Main success scenario**

1. User/Administrator clicks the ‘Stock Detail Update’ option in the homepage “START” menu bar.
2. System opens the stock details update form.
3. User/Administrator selects the critical input for which the stockbook entry is made.
4. System updates the selected critical input’s opening stock balance and its unit.
5. User/Administrator enters the necessary details and clicks “Save” button.
6. System saves the current stockbook details in the database.
7. System enters the information about user, domain and stock detail update in the transaction log.
8. System displays “Stockbook record updated successfully” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 5.a. | Invalid date of record: | Handle invalid date error. |
| 5.b. | Invalid field details: | Handle invalid field details. |
| 5.c. | Empty field: | Handle empty field error. |
| 5.d. | Incomplete form submission: | Handle incomplete form. |
| 6.a. | System fails to save the details | Handle save failure. |

**Special requirement**

To enter a record the user should enter a date which is after the last saved date in the stockbook otherwise the user cannot proceed further. This operation will be logged in the transaction log and the user will be held answerable for the operation.

**UC 2.2.2: Stock Detail Edit.** Using this use case the user can edit the last stockbook entry of a critical input.

*Primary actors:* User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface for editing the last stockbook entry of a critical input.

*Pre-condition*: The homepage is displayed with ‘Stock Detail Edit’ option under ‘Stock Details’ under ‘KVK Stockbook’, for editing the last stockbook entry of a critical input.

*Post-conditions*: The last stockbook entry of a critical input is edited with proper stockbook details.

**Main success scenario**

1. User clicks the ‘Stock Detail Edit’ option in the homepage “START” menu bar.
2. System opens the stock details edit form.
3. User/Administrator selects the critical input for editing its last stockbook entry.
4. System updates the form with last saved details of the selected critical input.
5. User/Administrator clicks “Edit” button for modifying field contents.
6. System converts the required read only fields as editable fields.
7. User/Administrator modifies the necessary details and clicks “Save” button.
8. System saves the edited stockbook details in the database.
9. System enters the information about user, domain and stock detail edit in the transaction log.
10. System displays “Stockbook record edited successfully” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 7.a. | Invalid field details: | Handle invalid field details. |
| 7.b. | Empty field: | Handle empty field error. |
| 7.c. | Incomplete form submission: | Handle incomplete form. |
| 8.a. | System fails to save the details | Handle save failure. |

**Special requirement**

The user can only edit the last stockbook record of the selected critical input. If any other record is to be edited then the records made after the record to be edited, must be deleted before continuing. The operations will be logged in the transaction log and the user will be held answerable for the operations.

**UC 2.2.3: Stock Detail Delete.** Using this use case the user can delete the last stockbook entry of a critical input.

*Primary actors:* User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface for deleting the last stockbook entry of a critical input.

*Pre-condition*: The homepage is displayed with ‘Stock Detail Delete’ option under ‘Stock Details’ under ‘KVK Stockbook’, for deleting the last stockbook entry of a critical input.

*Post-conditions*: The last stockbook entry of a critical input is deleted permanently.

**Main success scenario**

1. User/Administrator clicks the ‘Stock Detail Delete’ option in the homepage “START” menu bar.
2. System opens the stock details delete form.
3. User/Administrator selects the critical input for deleting its last stockbook entry.
4. System updates the form with last saved details of the selected critical input.
5. User/Administrator clicks “Delete” button for deleting the record.
6. System displays a delete confirmation message.
7. User/Administrator confirms record deletion.
8. System deletes the record from the database permanently.
9. System enters the information about user, domain and stock detail delete in the transaction log.
10. System displays “Stockbook record deleted successfully” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 5.a. | Incomplete form submission: | Handle incomplete form. |
| 8.a. | Failure to delete: | Handle failure to delete. |

**Special requirement**

The user must be careful while deleting a record as this type of operation will be logged in the transaction log and the user will be held answerable for the operation.

**UC 2.3: Stockbook Statement.** Using this use case the user/administrator can view the stockbook statement for a single critical input or all critical inputs for the selected time period and also take print out of the statement.

*Actors:* User, Administrator

**Stakeholders and interests**

*User*: Wants a user-friendly interface for viewing the stockbook statement.

*Administrator*: Wants a user-friendly interface for viewing and printing the stockbook statement.

*Pre-condition*: The homepage is displayed with ‘Stockbook Statement’ option under ‘KVK Stockbook’, for viewing/printing the stockbook statement.

*Post-conditions*: The stockbook statement is viewed/printed.

*Primary Actor*: Administrator

**Main success scenario**

1. Administrator clicks the ‘Stockbook Statement’ option in the homepage “START” menu bar.
2. System opens the stockbook statement form for viewing and printing.
3. Administrator selects the date range by clicking the date time picker buttons.
4. Administrator clicks “Show” button to view the stockbook statement.
5. System displays the stockbook statement of the selected date range.
6. Administrator clicks the “Print” button to print the statement.
7. System generates and shows a statement report ready for printing.
8. Administrator does necessary page setup and takes the print out.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 4.a. | Invalid date of range: | Handle invalid date range error. |
| 4.b. | Empty field: | Handle empty field error. |
| 4.c. | Incomplete form submission: | Handle incomplete form. |
| 5.a. | System fails to show statement: | Handle failure. |
| 7.a. | System fails to print statement: | Handle printing failure. |

**Special requirement**

The administrator must select a proper date range for viewing or printing the stockbook statement.

*Primary Actor*: User

**Alternate success scenario**

1. User clicks the ‘Stockbook Statement’ option in the homepage “START” menu bar.
2. System opens the stockbook statement form for viewing and printing.
3. User selects the date range by clicking the date time picker buttons.
4. User clicks “Show” button to view the stockbook statement.
5. System displays the stockbook statement of the selected date range.
6. User clicks the “Print” button to print the statement.
7. System checks whether the user is a member of the administrator domain.
8. System displays “User must be a member of administrator domain to continue” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 4.a. | Invalid date of range: | Handle invalid date range error. |
| 4.b. | Empty field: | Handle empty field error. |
| 4.c. | Incomplete form submission: | Handle incomplete form. |
| 5.a. | System fails to show statement: | Handle failure. |

**Special requirement**

The user must select a proper date range for viewing the stockbook statement. The user must be a member of the administrator domain to continue printing the statement.

**UC 3: Current Status.** Using this use case the user can view the current status of the stockbook i.e. the last successful stockbook entry of a critical input.

*Primary actors:* User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface for viewing the current status of stockbook for a critical input.

*Pre-condition*: The homepage is displayed with ‘Current Status’, for viewing the current status of a critical input.

*Post-conditions*: The current status of a critical input in the stockbook is viewed.

**Main success scenario**

1. User/Administrator clicks the ‘Current Status’ option in the homepage “START” menu bar.
2. System opens the stockbook current status form.
3. User/Administrator selects the critical input from the dropdown list.
4. System loads all the stockbook details of the selected critical input in the form.
5. User/Administrator views all the details and closes the form.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | System fails to show the details | Handle failure. |

**Special requirement**

The user/administrator must select a critical input from the dropdown list before continuing.

**UC 4: Transaction.** Using this use case the user/administrator can view all the transactions carried out on the software. Transaction types include Login Status, Password Change, New Domain Creation, Domain Name Deletion, New User ID Creation, User ID Deletion, New Critical Input, Critical Input Deletion, Backup Database, Restore Database, Scheduled Backup, Stock Detail Update, Stock Detail Edit and Stock Detail Delete.

*Primary actors:* User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface for viewing the transactions carried out on the software.

*Pre-condition*: The homepage is displayed with ‘Transactions’, for viewing the transactions.

*Post-conditions*: The transactions carried out on the software is viewed.

**Main success scenario**

1. User/Administrator clicks the ‘Transactions’ option in the homepage “START” menu bar.
2. System opens the transactions form.
3. User/Administrator selects the type of transaction to be viewed and clicks “Show” button.
4. System displays all the selected type of transactions in a tabular format.
5. User/Administrator views the transaction details and closes the form.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | System fails to show the details | Handle failure. |

**Special requirement**

The user/administrator must select a type of transaction form the dropdown list before continuing.

**UC 5: System Tool.** Using this use case the user can perform database.

**UC 5.1: Database Tasks.** Using this use case the administrator can perform database backup and database restore tasks.

**UC 5.1.1: Backup Database.** Using this use case the administrator can backup the database.

*Primary actor:* Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface for database backup.

*Pre-condition*: The homepage is displayed with ‘Backup Database’ under ‘Database Tasks’ under ‘System Tool’, for database backup.

*Post-conditions*: The database till the last transaction is backup successfully.

**Main success scenario**

1. Administrator clicks the ‘Backup Database’ option in the homepage “START” menu bar.
2. System opens ‘Browse for folder’ dialog.
3. Administrator selects the desired path and clicks “Ok” button.
4. System creates a copy of the current database in the selected path.
5. System enters the information about user, domain and backup database details in the transaction log.
6. System displays “Backup successful” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 4.a. | System fails to create backup | Handle backup failure. |

**Special requirement**

The administrator must select a valid path for database backup. This operation will be logged in the transaction log and the administrator will be held answerable for the operation.

*Primary actor*: User

**Alternate success scenario**

1. User clicks the ‘Backup Database’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to backup database.

**UC 5.1.2: Restore Database.** Using this use case the administrator can restore the database.

*Primary actor:* Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface for database restore.

*Pre-condition*: The homepage is displayed with ‘Restore Database’ under ‘Database Tasks’ under ‘System Tool’, for database restore.

*Post-conditions*: The current database’s state is restored to the selected database’s state successfully.

**Main success scenario**

1. Administrator clicks the ‘Restore Database’ option in the homepage “START” menu bar.
2. System displays “Please save all unsaved information before continuing” message.
3. System asks for database restore confirmation.
4. System displays “Select KVKDhalai\_Stockbook.mdf file.” message.
5. System opens ‘Open file’ dialog.
6. Administrator selects ‘KVKDhalai\_Stockbook.mdf’ file and clicks “Open” button.
7. System copies the ‘KVKDhalai\_Stockbook.mdf’ file to a temporary location.
8. System displays “Select KVKDhalai\_Stockbook\_log.ldf file.” message.
9. System opens ‘Open file’ dialog.
10. Administrator selects ‘KVKDhalai\_Stockbook\_log.ldf’ file and clicks “Open” button.
11. System copies the ‘KVKDhalai\_Stockbook\_log.ldf’ file to a temporary location.
12. System displays “Please restart application IMMEDIATELY to continue.” message.
13. System enters ‘Database Restore’ ‘Partially successful’ in the transaction log.
14. System closes the stockbook software automatically.
15. Administrator reopens the stockbook software.
16. System checks the transaction log for any record containing ‘Restore Database’ ‘Partially successful’.
17. System copies both ‘KVKDhalai\_Stockbook.mdf’ and ‘KVKDhalai\_Stockbook\_log .ldf’ file from the temporary location to the application directory.
18. System enters ‘Database Restore’ ‘Successful’ in the transaction log.
19. System displays “Database restore successful.” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 6.a. | Administrator selects invalid file: | Handle failure due to invalid file. |
| 7.a. | System fails to copy file: | Handle copy failure |
| 10.a. | Administrator selects invalid file: | Handle failure due to invalid file. |
| 11.a. | System fails to copy file: | Handle copy failure |
| 17.a. | System fails to copy file: | Handle copy failure |

**Special requirement**

The administrator must select the valid files that are asked during database restore. This operation will be logged in the transaction log and the administrator will be held answerable for the operation.

*Primary actor*: User

**Alternate success scenario**

1. User clicks the ‘Restore Database’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to restore database.

**UC 5.2: User Accounts Task.** Using this use case the user/administrator can manage user accounts and domains.

**UC 5.2.1: Change Password.** Using this use case the user/administrator can change their respective password.

*Primary actors*: User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface to change password.

*Pre-condition*: The homepage is displayed with ‘Change Password’ option under ‘User Accounts Task’ under ‘System Tool’ in the homepage “START” menu bar.

*Post-conditions*: The password change operation is completed successfully.

**Main success scenario**

1. User/Administrator clicks ‘Change Password’ in the homepage “START” menu bar.
2. System displays a form for changing the login password.
3. User/Administrator enters the required details and clicks “Save” button.
4. System validates the old password and saves the new password.
5. System enters the information about user, domain and password change information in the transaction log.
6. System displays ‘Password Change successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | Invalid old password: | Handle invalid old password. |
| 4.b. | Failure to save password: | Handle failure to save new password. |

**UC 5.2.2: New User Account.** Using this use case the administrator can create a new user account under the existing domain(s) or under a new domain.

*Primary actor*: Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface to create a new user account under the existing domain(s) or under a new domain.

*Pre-condition*: The homepage is displayed with ‘New User Account’ option under ‘User Accounts Task’ under ‘System Tool’ in the homepage “START” menu bar.

*Post-conditions*: The user account created successfully.

**Main success scenario**

1. Administrator clicks ‘New User Account’ in the homepage “START” menu bar.
2. System displays a form for creating a new user account.
3. Administrator selects an existing domain from the drop down list.
4. Administrator enters the required details and clicks “Create” button.
5. System validates the details and saves the new user account details in database.
6. System enters the information about user, domain and new user id creation information in the transaction log.
7. System displays ‘New user created successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 4.a. | Invalid user account details: | Handle invalid details error. |
| 4.b. | User ID already exist: | Handle already exist error. |
| 4.c. | Empty field: | Handle empty field error. |
| 4.d. | Incomplete form submission: | Handle incomplete form. |
| 5.a. | Failure to save details: | Handle failure to save. |

**Special requirement**

The administrator must have valid reasons for creating the new user account because this operation will be logged in the transaction log and the administrator will be held answerable for the operation

*Primary actor*: Administrator

**Alternate success scenario 1**

1. Administrator clicks ‘New User Account’ in the homepage “START” menu bar.
2. System displays a form for creating a new user account.
3. Administrator selects “Create New” domain option from the drop down list.
4. System displays a field to enter the new domain name.
5. Administrator enters the required details and clicks “Create” button.
6. System validates the details and saves the new user account details in database.
7. System enters the information about user, domain and new user id creation information in the transaction log.
8. System enters the information about user, domain and new domain creation information in the transaction log.
9. System displays ‘New user created successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 5.a. | Invalid user account details: | Handle invalid details error. |
| 5.b. | Domain name already exist: | Handle already exist error. |
| 5.c. | User ID already exist: | Handle already exist error. |
| 5.d. | Empty field: | Handle empty field error. |
| 5.e. | Incomplete form submission: | Handle incomplete form. |
| 6.a. | Failure to save details: | Handle failure to save. |

**Special requirement**

The administrator must have valid reasons for creating the new user account and a new domain because this operation will be logged in the transaction log and the administrator will be held answerable for the operation

*Primary actor*: User

**Alternate success scenario 2**

1. User clicks the ‘New User Account’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to backup database.

**UC 5.2.3: Delete User Account.** Using this use case the administrator can delete any user account from any domain.

*Primary actor*: Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface to delete a user account under the existing domain(s).

*Pre-condition*: The homepage is displayed with ‘Delete User Account’ option under ‘User Accounts Task’ under ‘System Tool’ in the homepage “START” menu bar.

*Post-conditions*: The user account deleted successfully.

**Main success scenario**

1. Administrator clicks ‘Delete User Account’ in the homepage “START” menu bar.
2. System displays a form for deleting a user account.
3. Administrator selects an existing domain from the drop down list.
4. System loads all the user id of the selected domain in a drop down list.
5. Administrator selects a user id from the drop down list and clicks “Delete” button.
6. System asks for a confirmation “Confirm user account deletion”.
7. Administrator clicks “Yes” button.
8. System deletes the selected user id from the selected domain.
9. System enters the information about user, domain and user id deletion information in the transaction log.
10. System displays “User ID deleted successfully” message.
11. Administrator clicks “Cancel” button.
12. System closes the form for deleting user id.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 5.a. | Incomplete form submission: | Handle incomplete form. |
| 8.a. | Failure to delete: | Handle failure to delete. |

**Special requirement**

The administrator must have valid reasons for deleting the user account because this operation will be logged in the transaction log and the administrator will be held answerable for the operation.

*Primary actor*: User

**Alternate success scenario 1**

1. User clicks the ‘Delete User Account’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to delete user account.

**UC 5.2.4: Delete Domain.** Using this use case the administrator can delete any existing domain(s) except the administrator domain.

*Primary actor*: Administrator

**Stakeholders and interests**

*Administrator*: Wants a user-friendly interface to delete a domain.

*Pre-condition*: The homepage is displayed with ‘Delete Domain’ option under ‘User Accounts Task’ under ‘System Tool’ in the homepage “START” menu bar.

*Post-conditions*: The domain is deleted successfully.

**Main success scenario**

1. Administrator clicks ‘Delete Domain’ in the homepage “START” menu bar.
2. System displays a form for deleting a domain.
3. Administrator selects an existing domain from the drop down list and clicks “Delete” button.
4. System asks for a confirmation “Confirm domain name deletion”.
5. Administrator clicks “Yes” button.
6. System deletes the selected domain.
7. System enters the information about user, domain and domain name deletion information in the transaction log.
8. System displays “Domain name deleted successfully” message.
9. Administrator clicks “Cancel” button.
10. System closes the form for deleting domain.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 6.a. | Failure to delete: | Handle failure to delete. |

**Special requirement**

The administrator must be careful while deleting a domain as this will lead to deletion of all user accounts under this domain. The administrator must have valid reasons for deleting the domain because this operation will be logged in the transaction log and the administrator will be held answerable for the operation.

*Primary actor*: User

**Alternate success scenario 1**

1. User clicks the ‘Delete Domain’ option in the homepage “START” menu bar.
2. System checks whether the user is a member of the administrator domain.
3. System displays “User must be a member of administrator domain to continue” message.

**Special requirement**

The user must be a member of the administrator domain to delete domain.

**UC 6: About.** Using this use case the user can view the details of the softwares build and developer contact information.

*Primary actors*: User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a user-friendly interface to view the software details.

*Pre-condition*: The homepage is displayed with ‘About’ option in the homepage “START” menu bar.

*Post-conditions*: The required software details are displayed in a well arranged format.

**Main success scenario**

1. User/Administrator clicks ‘About’ in the homepage “START” menu bar.
2. System displays a user-friendly form containing different tabs with different information regarding the software and the developer.
3. User/Administrator views all the information and closes the form.

**UC 7: Logout.** Using this use case the user can logout the current session. All unsaved open forms will be locked. No use of the software is possible before login.

*Primary actors*: User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants a logout option for sudden pause in work with unsaved details.

*Pre-condition*: The homepage is displayed with ‘Logout’ option in the homepage “START” menu bar.

*Post-conditions*: The user/administrator is logged out of the software.

**Main success scenario**

1. User/Administrator clicks ‘Logout’ in the homepage “START” menu bar.
2. System logs out the user from the software.
3. System locks access to all the open forms.
4. System displays a login form.

**Alternate success scenario**

1. User/Administrator clicks ‘Logout’ in the context menu.
2. System logs out the user from the software.
3. System locks access to all the open forms.
4. System displays a login form.

**UC 8: Exit.** Using this use case the user/administrator exits from the software. All unsaved open forms will be closed and the unsaved information will be lost.

*Primary actors*: User, Administrator

**Stakeholders and interests**

*User/Administrator*: Wants to exit from the successfully.

*Pre-condition*: The homepage is displayed with ‘Exit’ option in the homepage “START” menu bar.

*Post-conditions*: The user/administrator is logged out of the software.

**Main success scenario**

1. User/Administrator clicks ‘Exit’ in the homepage “START” menu bar.
2. System asks for a confirmation “Do you want to exit?”
3. User/Administrator clicks “Yes” button.
4. System exits from the software.

**Special requirement**

The user/administrator must be careful while exit operation as all the unsaved stockbook information will be lost. It is advisable to save and close all the open forms before clicking ‘Exit’.