Text Description:-

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

*Primary actor:* User

**Stakeholders and interests**

*User*: Wants a user-friendly environment and proper steps for maintaining the office cash book.

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

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

**Main success scenario**

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

**Extensions**

|  |  |  |
| --- | --- | --- |
| 2.a. | Invalid username or password: | Handle invalid username or password error. |
| 2.b. | Empty field: | Handle empty field error. |
| 2.c. | Incomplete form submission: | Handle incomplete form. |
| 3.a. | Invalid login details: | Handle invalid login details. |
|  |  |  |

**Alternate success scenario**

1. System displays a login form.
2. User enters username and password.
3. User checks the ‘Change login details’ option and clicks “Login”.
4. System validates the username and password and logs into the software.
5. System enters the 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.
8. System displays a form for changing the login password.
9. User enters the required details and clicks “Save”.
10. System validates the old password and saves the new password.
11. System enters the password change information in the transaction log.
12. System displays ‘Password Change successful’ message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 2.a. | Invalid username or password: | Handle invalid username or password error. |
| 2.b. | Empty field: | Handle empty field error. |
| 3.a. | Incomplete form submission: | Handle incomplete form. |
| 4.a. | Invalid login details: | Handle invalid login details. |
| 9.a. | Incomplete form submission: | Handle incomplete form. |
| 10.a. | Invalid old password: | Handle invalid old password. |
| 10.b. | Failure to save password: | Handle failure to save new password. |

**UC 2: KVK Cashbook.** Using this use case the user can manage the cash book records and also take case book statements for a selected time period.

**UC 2.1: Cash Details.** Using this use case the user can manage the cash book details like Update, Edit and Delete cash book records.

**UC 2.1.1: Cash Detail Update.** Using this use case the user can create a new cash book entry for a day.

*Primary actor:* User

**Stakeholders and interests**

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

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

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

**Main success scenario**

1. User clicks the ‘Cash Detail Update’ option in the homepage “START” menu bar.
2. System opens the receipts and payments entry form.
3. User enters the necessary details in the receipts section and clicks “Calculate” button in the receipts section.
4. System updates the opening balance and the grand total from the cashbook.
5. User enters the necessary details in the payments section and clicks “Calculate” button in the payments section.
6. System updates the closing balance and the grand total from the cashbook.
7. User clicks “Save” button.
8. System saves the current cashbook details in the database.
9. System enters the cash detail update information in the transaction log.
10. System displays “Cashbook record updated successfully” message.

**Extensions**

|  |  |  |
| --- | --- | --- |
| 3.a. | Invalid date of record: | Handle invalid date error. |
| 3.b. | Invalid field details: | Handle invalid field details. |
| 3.c. | Empty field: | Handle empty field error. |
| 5.a. | Invalid field details: | Handle invalid field details. |
| 5.b. | Empty field: | Handle empty field error. |
| 7.a. | Incomplete form submission: | Handle incomplete form. |
| 8.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 cashbook 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 2.1.2: Cash Detail Edit.** Using this use case the user can edit the last cash book entry.

*Primary actor:* User

**Stakeholders and interests**

*User*: Wants a user-friendly interface for editing the last cashbook entry.

*Pre-condition*: The homepage is displayed with ‘Cash Detail Edit’ option under ‘Cash Details’ under ‘KVK Cashbook’, for editing the last cashbook entry.

*Post-conditions*: The last cashbook entry is edited with proper cashbook details.

**Main success scenario**

1. User clicks the ‘Cash Detail Edit’ option in the homepage “START” menu bar.
2. System opens the receipts and payments entry form displaying the last entered data.
3. User clicks “Edit” button for modifying field contents.
4. System converts the required read only fields as editable fields.
5. User edits the necessary details in the receipts section and clicks “Calculate” button in the receipts section.
6. System updates the opening balance and the grand total from the cashbook.
7. User edits the necessary details in the payments section and clicks “Calculate” button in the payments section.
8. System updates the closing balance and the grand total from the cashbook.
9. User clicks “Save” button.
10. System saves the edited cashbook details in the database.
11. System enters the cash detail edit information in the transaction log.
12. System displays “Cashbook record edited 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. |
| 7.a. | Invalid field details: | Handle invalid field details. |
| 7.b. | Empty field: | Handle empty field error. |
| 9.a. | Incomplete form submission: | Handle incomplete form. |
| 10.a. | System fails to save the details | Handle save failure. |

**Special requirement**

The user can only edit the last cashbook record. 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 2.1.3: Cash Detail Delete.** Using this use case the user can delete the last cash book entry.

*Primary actor:* User

**Stakeholders and interests**

*User*: Wants a user-friendly interface for deleting the last cashbook entry.

*Pre-condition*: The homepage is displayed with ‘Cash Detail Delete’ option under ‘Cash Details’ under ‘KVK Cashbook’, for deleting the last cashbook entry.

*Post-conditions*: The last cashbook entry is deleted permanently.

**Main success scenario**

1. User clicks the ‘Cash Detail Delete’ option in the homepage “START” menu bar.
2. System opens the receipts and payments entry form displaying the last entered data.
3. User clicks “Delete” button for deleting the record.
4. System displays a delete confirmation message.
5. User confirms record deletion.
6. System deletes the record from the database.
7. System enters the cash detail delete information in the transaction log.
8. System displays “Cashbook record deleted successfully” message.

**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 2.2: Cashbook Statement.** Using this use case the user can view the cashbook statement of the selected time period and also take print out of the statement.

*Primary actor:* User

**Stakeholders and interests**

*User*: Wants a user-friendly interface for viewing and printing the cashbook statement.

*Pre-condition*: The homepage is displayed with ‘Cashbook Statement’ option under ‘KVK Cashbook’, for viewing and printing the cashbook statement.

*Post-conditions*: The cashbook statement is viewed and printed.

**Main success scenario**

1. User clicks the ‘Cashbook Statement’ option in the homepage “START” menu bar.
2. System opens the cashbook 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 cashbook statement.
5. System displays the cashbook statement of the selected date range.
6. User clicks the “Print” button to print the statement.
7. System generates and shows a statement report ready for printing.
8. User 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. |
| 7.a. | System fails to print statement: | Handle printing failure. |

**Special requirement**

The user must select a proper date range for viewing or printing the cashbook statement.

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

*Primary actor:* User

**Stakeholders and interests**

*User*: Wants a user-friendly interface for viewing the current status of cashbook.

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

*Post-conditions*: The current status of cashbook is viewed.

**Main success scenario**

1. User clicks the ‘Current Status’ option in the homepage “START” menu bar.
2. System opens the cashbook current status form containing the last cashbook entry.
3. User views all the details and closes the form.

**UC 4: Transaction.** Using this use case the user can view all the transactions carried out on the software. Transaction types include Login Status, Password Change, Backup Database, Restore Database, Scheduled Backup, Cash Detail Update, Cash Detail Edit and Cash Detail Delete.

*Primary actor:* User

**Stakeholders and interests**

*User*: 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 clicks the ‘Transactions’ option in the homepage “START” menu bar.
2. System opens the transactions form.
3. User 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 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 must select a type of transaction form the list before continuing.

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

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

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

*Primary actor:* User

**Stakeholders and interests**

*User*: 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. User clicks the ‘Backup Database’ option in the homepage “START” menu bar.
2. System opens ‘Browse for folder’ dialog.
3. User 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 backup database information in the transaction log.
6. System displays “Backup successful” message.

**Extensions**

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

**Special requirement**

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

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

*Primary actor:* User

**Stakeholders and interests**

*User*: 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. User 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\_Cashbook.mdf file.” message.
5. System opens ‘Open file’ dialog.
6. User selects ‘KVKDhalai\_Cashbook.mdf’ file and clicks “Open” button.
7. System copies the ‘KVKDhalai\_Cashbook.mdf’ file to a temporary location.
8. System displays “Select KVKDhalai\_Cashbook\_log.ldf file.” message.
9. System opens ‘Open file’ dialog.
10. User selects ‘KVKDhalai\_Cashbook\_log.ldf’ file and clicks “Open” button.
11. System copies the ‘KVKDhalai\_Cashbook\_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 cashbook software automatically.
15. User reopens the cashbook software.
16. System checks the transaction log for any record containing ‘Restore Database’ ‘Partially successful’.
17. System copies both ‘KVKDhalai\_Cashbook.mdf’ and ‘KVKDhalai\_Cashbook\_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. | User selects invalid file: | Handle failure due to invalid file. |
| 7.a. | System fails to copy file: | Handle copy failure |
| 10.a. | User 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 user must select the valid files that are asked during database restore. This operation will be logged in the transaction log and the user will be held answerable for the operation.

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

*Primary actor*: User

**Stakeholders and interests**

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

*Pre-condition*: The homepage is displayed with ‘About’ option to view the software details.

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

**Main success scenario**

1. User 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 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 actor*: User

**Stakeholders and interests**

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

*Pre-condition*: The homepage is displayed with ‘Logout’ option to logout from the software.

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

**Main success scenario**

1. User 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 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*: 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 is logged out of the software.

**Main success scenario**

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

**Special requirement**

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