

**پروژه درس مهندسی نرم افزار**

**Use Case Description**

**استاد:**

دکتر محمودزاده

**اعضای گروه:**

صادق اکبری

عرفان رئوفیان

علی غفورزاده

سید مهدی غضنفری

امیرعباس مهدی زاده

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: login | ID: 1 | | Importance Level: High |
| Primary Actor: User | | Use Case Type: | |
| Stakeholders and Interests: User - Wants to log in. | | | |
| Brief Description: This use case describes how the user can log into his account. | | | |
| Trigger: User clicks the login button  Type: External | | | |
| Relationships:  Association: User  Include:  Extend:  Generalization: | | | |
| Normal Flow of Events:   1. Enter username 2. Enter password 3. User is logged in | | | |
| Sub Flows: | | | |
| Alternative/Exceptional Flows:  If the user enters the wrong password:   1. User clicks account recovery button 2. If password incorrect for multiple times: force user to recover the account 3. Else: go to the normal flow | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Sign Up | ID: 2 | | Importance Level: High |
| Primary Actor: User | | Use Case Type: | |
| Stakeholders and Interests: User – wants to create an account | | | |
| Brief Description: This use case describes how the user can create an account | | | |
| Trigger: user clicks the sign-up button  Type: External | | | |
| Relationships:  Association: User  Include:  Extend:  Generalization: | | | |
| Normal Flow of Events:   1. Enters username 2. Enters password 3. Clicks the “done” button 4. Sees successful message | | | |
| Sub Flows: | | | |
| Alternative/Exceptional Flows:   1. If username not unique:  Enters another username again and goes to the normal flow of use case 2. If password not strong:  Enters another password again and goes to the normal flow of use case | | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: financial account management | ID: 3 | Importance Level: very high |
| Primary Actor: User | Use Case Type: | |
| Stakeholders and Interests: User – wants to manage his financial account | | |
| Brief Description: This use case describes how the user can manage his financial account | | |
| Trigger: login process triggers this  Type: Internal | | |
| Relationships:  Association: User  Include: Add new financial account and category, delete the financial account, edit financial account and category.  Extend:  Generalization: | | |
| Normal Flow of Events:  User can either click:   * Add a new financial account * Delete financial account * Edit financial account * Add new category * Edit category * Delete category | | |
| Sub Flows:   1. Add a new financial account:    1. Enters the name of the account    2. Get account balance    3. Clicks on create account 2. Delete financial account:    1. Select financial account    2. Clicks on delete financial account    3. User confirms 3. Edit financial account:    1. Select financial account    2. Clicks edit account    3. Changes old data    4. Clicks done 4. Add a new category:    1. Enters category name    2. Selects category label 5. Edit category    1. Selects a category previously created    2. Clicks on the edit button    3. Changes the data 6. Delete category    1. Selects a category previously created    2. Clicks on the delete button    3. User confirms | | |
| Alternative/Exceptional Flows:   1. Add a new financial account:    1. User clicks the button “account is a real bank account”    2. Enters real bank account number 2. Delete financial account:    1. User doesn’t confirm    2. The account won’t be deleted 3. Add a new category:    1. If the category name user entered already exists system shows an error message 4. Delete category    1. User doesn’t confirm    2. The category won’t be deleted | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: manage financial reports | ID: 4 | Importance Level: High |
| Primary Actor: User | Use Case Type: | |
| Stakeholders and Interests: User – wants to create a report from financial activities | | |
| Brief Description: This use case describes how the user can see a report of financial activities | | |
| Trigger: user clicks the reports button  Type: External | | |
| Relationships:  Association: User  Include:  Extend:  Generalization: | | |
| Normal Flow of Events:   1. Select one of the existing financial accounts 2. Filter categories 3. Select timeline 4. Sees financial report of this account in several statistic forms | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows:   1. If the selected account has multiple users:  Reports have also information about other users 2. Account has no information considered the applied filters by user:  User sees empty information errors | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: transactions management | ID: 5 | Importance Level: very high |
| Primary Actor: User | Use Case Type: | |
| Stakeholders and Interests: User – wants to manage his transactions | | |
| Brief Description: This use case describes how the user can manage his transactions | | |
| Trigger: user clicks “transactions” button  Type: external | | |
| Relationships:  Association: User  Include: Add income/outcome/debt/loan, edit income/outcome/debt/loan  Extend:  Generalization: | | |
| Normal Flow of Events:  User can either click:   * Add new income * Edit income * Add new outcome * Edit outcome * Add new debt * Edit debt * add loan * edit loan | | |
| Sub Flows:  1. Add new income:   * 1. Select category   2. Enter amount   3. Select date and time   4. Select related account   5. Submit new income  1. Edit income:    1. Select one of the incomes    2. Edit amount    3. Edit category    4. Edit date and time    5. Change related account    6. Submit changes 2. Add a new outcome:    1. Select category    2. Enter amount    3. Select date and time    4. Select related account    5. Submit a new outcome 3. Edit outcome:    1. Select one of the outcomes    2. Edit category    3. Enter new amount    4. Select date and time    5. Change related account    6. Submit changes 4. Add new debt    1. Enter creditor name    2. Enter amount    3. Enter description    4. Select the date of settlement    5. Submit new debt 5. edit debt    1. Select one of the debts    2. Enter new creditor name    3. Enter new amount    4. Enter new description    5. Change date of settlement    6. Submit changes | | |
| Alternative/Exceptional Flows: | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: financial group management | ID: 6 | Importance Level: medium |
| Primary Actor: Group Admin | Use Case Type: | |
| Stakeholders and Interests: Group Admin – wants to manage his financial group | | |
| Brief Description: This use case describes how the user can manage his financial group | | |
| Trigger: group admin clicks financial group button  Type: External | | |
| Relationships:  Association: Group admin  Include: Add new financial group, delete a financial group, edit a financial group.  Extend:  Generalization: | | |
| Normal Flow of Events:  User can either click:   * Add new financial group * Delete financial group * Add a member to the financial group * Delete member(s) from the financial group | | |
| Sub Flows:  1. Add a new financial group:   * 1. Enters the name of the financial group   2. Chooses members list   3. Clicks on add contact(s)   4. Clicks on create a financial group  1. Delete a financial group:    1. Select financial group    2. Clicks on delete financial group    3. User confirms 2. Edit financial group:    1. Select financial group    2. Clicks edit financial group    3. Changes old data    4. Clicks done 3. Add a member to a financial group:    1. Select financial group    2. Clicks financial group members    3. Clicks add a group member    4. Choose group member    5. Clicks done    6. Clicks save changes    7. User confirms 4. Delete a member from the financial group:    1. Select financial group    2. Clicks edit financial group    3. Clicks financial group members    4. Choose group member    5. Clicks delete group member    6. Clicks done | | |
| Alternative/Exceptional Flows:   1. Add new contact:    1. If the contact name or phone number user entered already exists system shows an error message 2. Delete contact:    1. User doesn’t confirm    2. The contact won’t be deleted | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: contact management | ID: 7 | Importance Level: medium |
| Primary Actor: User | Use Case Type: | |
| Stakeholders and Interests: User – wants to manage his contacts | | |
| Brief Description: This use case describes how the user can manage his contacts | | |
| Trigger: user clicks contacts button  Type: External | | |
| Relationships:  Association: User  Include: Add new contact, delete a contact, edit a contact.  Extend:  Generalization: | | |
| Normal Flow of Events:  User can either click:   * Add new contact * Delete contact * Edit contact | | |
| Sub Flows:  1. Add new contact:   * 1. Enters the name of the contact   2. Enters the phone of contact   3. Clicks on create contact  1. Delete contact:    1. Select contact    2. Clicks on delete contact    3. User confirms 2. Edit contact:    1. Select contact    2. Clicks edit contact    3. Changes old data    4. Clicks done | | |
| Alternative/Exceptional Flows:   1. Add new contact:    1. If the contact name or phone number user entered already exists system shows an error message 2. Delete contact:    1. User doesn’t confirm    2. The contact won’t be deleted | | |