



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: <ul style="list-style-type: none">• 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: 1.1. Enters the name of the account 1.2. Get account balance 1.3. Clicks on create account 2. Delete financial account: 2.1. Select financial account 2.2. Clicks on delete financial account 2.3. User confirms 3. Edit financial account: 3.1. Select financial account 3.2. Clicks edit account 3.3. Changes old data 3.4. Clicks done 4. Add a new category: 4.1. Enters category name		

4.2. Selects category label

5. Edit category

5.1. Selects a category previously created

5.2. Clicks on the edit button

5.3. Changes the data

6. Delete category

6.1. Selects a category previously created

6.2. Clicks on the delete button

6.3. User confirms

Alternative/Exceptional Flows:

1. Add a new financial account:

1.1. User clicks the button "account is a real bank account"

1.2. Enters real bank account number

2. Delete financial account:

2.1. User doesn't confirm

2.2. The account won't be deleted

3. Add a new category:

3.1. If the category name user entered already exists system shows an error message

4. Delete category

4.1. User doesn't confirm

4.2. 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: <ul style="list-style-type: none">• 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: 1.1. Select category 1.2. Enter amount 1.3. Select date and time 1.4. Select related account 1.5. Submit new income 2. Edit income: 2.1. Select one of the incomes 2.2. Edit amount 2.3. Edit category 2.4. Edit date and time 2.5. Change related account 2.6. Submit changes 3. Add a new outcome: 3.1. Select category 3.2. Enter amount 3.3. Select date and time 3.4. Select related account 3.5. Submit a new outcome		

4. Edit outcome:
 - 4.1. Select one of the outcomes
 - 4.2. Edit category
 - 4.3. Enter new amount
 - 4.4. Select date and time
 - 4.5. Change related account
 - 4.6. Submit changes
5. Add new debt
 - 5.1. Enter creditor name
 - 5.2. Enter amount
 - 5.3. Enter description
 - 5.4. Select the date of settlement
 - 5.5. Submit new debt
6. edit debt
 - 6.1. Select one of the debts
 - 6.2. Enter new creditor name
 - 6.3. Enter new amount
 - 6.4. Enter new description
 - 6.5. Change date of settlement
 - 6.6. Submit changes

Alternative/Exceptional Flows:

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: <ul style="list-style-type: none"> • Add new financial group • Delete financial group • Add a member to the financial group • Delete member(s) from the financial group 	
Sub Flows: <ol style="list-style-type: none"> 1. Add a new financial group: <ol style="list-style-type: none"> 1.1. Enters the name of the financial group 1.2. Enters notification settings 1.3. Chooses members list 1.4. Sets association percentage for each user 1.5. Enters limit for each user 1.6. Sets notification for each user 1.7. Clicks on add contact(s) 1.8. Clicks on create a financial group 2. Delete a financial group: <ol style="list-style-type: none"> 2.1. Select financial group 2.2. Clicks on delete financial group 2.3. User confirms 3. Edit financial group: <ol style="list-style-type: none"> 3.1. Select financial group 3.2. Clicks edit financial group 3.3. Changes old data 3.4. Clicks done 4. Add a member to a financial group: <ol style="list-style-type: none"> 4.1. Select financial group 4.2. Clicks financial group members 4.3. Clicks add a group member 4.4. Choose group member 4.5. Enters association percentage 	

- 4.6. Enters limits
- 4.7. Sets notification settings
- 4.8. Clicks done
- 4.9. Clicks save changes
- 4.10. User confirms

5. Delete a member from the financial group:

- 5.1. Select financial group
- 5.2. Clicks edit financial group
- 5.3. Clicks financial group members
- 5.4. Choose group member
- 5.5. Clicks delete group member
- 5.6. Clicks done
- 5.7.

Alternative/Exceptional Flows:

7. Add new contact:

- 7.1. If the contact name or phone number user entered already exists system shows an error message

8. Delete contact:

- 8.1. User doesn't confirm
- 8.2. 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: <ul style="list-style-type: none">• Add new contact• Delete contact• Edit contact		
Sub Flows: 1. Add new contact: 5.8. Enters the name of the contact 5.9. Enters the phone of contact 5.10. Clicks on create contact 6. Delete contact: 6.1. Select contact 6.2. Clicks on delete contact 6.3. User confirms 7. Edit contact: 7.1. Select contact 7.2. Clicks edit contact 7.3. Changes old data 7.4. Clicks done		
Alternative/Exceptional Flows: 9. Add new contact:		

9.1. If the contact name or phone number user entered already exists system shows an error message

10. Delete contact:

10.1. User doesn't confirm

10.2. The contact won't be deleted