BNA Online Optimization

Requirements Determination and Use Case Analysis / System Proposal / Analysis Phase   
(Homework No.2)

Project team: Team 10

Instructor: Dr. Araz Yusubov

Submitted in partial fulfillment of the requirements of the INFT 2303: Systems Analysis and Design course project

|  |  |
| --- | --- |
| GitHub repository | https://github.com/ADA-SITE-INFT2303-2022-Spring/systems-development-project-team-10 |
| Version date | Version information |
| 23.03.2022 | Initial draft |
| 26.03.2022 | Final draft |
| 20.04.2022 | Version 2.0 |

|  |  |
| --- | --- |
| Other documents in the package | |
| File name | Brief description of the document |
| <File name> | <Description of the document> |
| <File name> | <Description of the document> |

|  |  |  |
| --- | --- | --- |
| Team member | Contribution to this homework (NOT the project) | Estimated % |
| Kanan Badalov | Use Case Analysis | 25 % |
| Farhad Khidirli | Use Case Analysis | 25 % |
| Kanan Ibrahimli | Requirements Definition - Function | 25 % |
| Bakhtiyar Guluzade | Requirements Definition | 25 % |

# Table of Contents

[Table of Contents 2](#_Toc140420295)

[1. Introduction 3](#_Toc2055915558)

[Definitions 3](#_Toc2048217310)

[2. Requirements Definition 3](#_Toc1063007619)

[Functional Requirements 3](#_Toc518470651)

[Nonfunctional Requirements 4](#_Toc1337108851)

[Design Constraints 4](#_Toc1393536096)

[3. Use Case Analysis 5](#_Toc1226744998)

[External Actor Descriptions 5](#_Toc1837400853)

[User Story Descriptions 5](#_Toc1642752093)

[Use Case Diagram 5](#_Toc2107774555)

[Use Case Descriptions 6](#_Toc991296628)

[Use Case 1 6](#_Toc429801668)

[Use Case 2 7](#_Toc571711769)

[Use Case 3 7](#_Toc840920768)

[Use Case 4 8](#_Toc1948796761)

[Use Case 5 9](#_Toc336254190)

[Use Case 6 9](#_Toc732233279)

[Use Case 7 10](#_Toc1940341300)

[Use Case 8 10](#_Toc1103172322)

[Use Case 9 11](#_Toc524602675)

[Use Case 10 12](#_Toc2072440254)

[Use Case 11 12](#_Toc1778439590)

[Use Case 12 12](#_Toc1163907255)

[4. References 13](#_Toc347321925)

# Introduction

**This is part of the System Proposal for a hypothetical project BNA Optimization System submitted for partial fulfillment of the requirements of the Systems Analysis and Design course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.**

**The content of the document includes requirements definition and use case analysis.**

## Definitions

|  |  |
| --- | --- |
| Term | Definition |
| Baku Card  Wireframes  Mockups | Baku Card is a usual card needed for paying public transport fare in Azerbaijan.  two-dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of content  model or replica of a machine or structure, used for instructional or experimental purposes. |
| OS  Apple Pay  Google Pay  Google Play  App Store  AZN | An Operating System (OS) is an interface between a computer user and computer hardware.  Apple Pay is a mobile payment and digital wallet service by Apple Inc. that allows users to make payments in person, in iOS apps, and on the web using Safari. It is supported on the iPhone, Apple Watch, iPad, and Mac.  Google Pay is a digital wallet platform and online payment system developed by Google to power in-app, online, and in-person contactless purchases on mobile devices, enabling users to make payments with Android phones, tablets, or watches.  Google Play, also branded as the Google Play Store and formerly Android Market, is a digital distribution service operated and developed by Google for Android OS.  An app store is a type of digital distribution platform for computer software called applications, often in a mobile context for IOS.  AZN is a local currency in Azerbaijan. |

# Requirements Definition

## Functional Requirements

**Process-oriented**

1. The new coming user shall be able to register to the system by filling out username, phone number, and password fields.
2. The user shall be able to sign into the system using username and password information.
3. The user shall be able to view profile information.
4. The user shall be able to edit profile information.
5. The user shall be able to add the Baku Card to his account.
6. The user shall be able to delete his card from his account.
7. The user shall be able to replenish his card.
8. The user shall be able to see the card’s balance.

**Information-oriented**

1. The system shall save the user information such as username, phone number, and password.
2. The system shall save the card information.
3. The system shall save changes by the user.
4. The system should save online payment transactions history for at least 3 months.

## Nonfunctional Requirements

**Performance Requirements**

The system should be able to handle 10 000 users simultaneously without performance deterioration. Although this is less than the number of “Baku Card” users, we should make it affordable for shareholders or the government. Because we assume not all users will enter our system at the same time. But the system needs to hold at least 1 million user information.

**Security Requirements**

The system needs to secure user privacy by making sure there is no leakage of crucial data. Moreover, data must be guarded against unauthorized access.

**Operational Requirements**

The system shouldn’t freeze in the middle of an action. Any transactions shall not take longer than 3 seconds. Any wrong action should not make the system shut down. Interface should be

straightforward and simple to use.

## Design Constraints

UX design (Background, Buttons, Wireframes) of the system needs to be coherent. We can use “Baku Card” color (blue, red, white) here for better understanding. Screen height and length ratios must be changed, depending on every device. Because Android and IOS have different working principles and screen sizes, the wireframe design should not cross limits. The app needs to work in minimum requirements for every device. A dark theme is also needed for users that like the black backgrounds.

# Use Case Analysis

## External Actor Descriptions

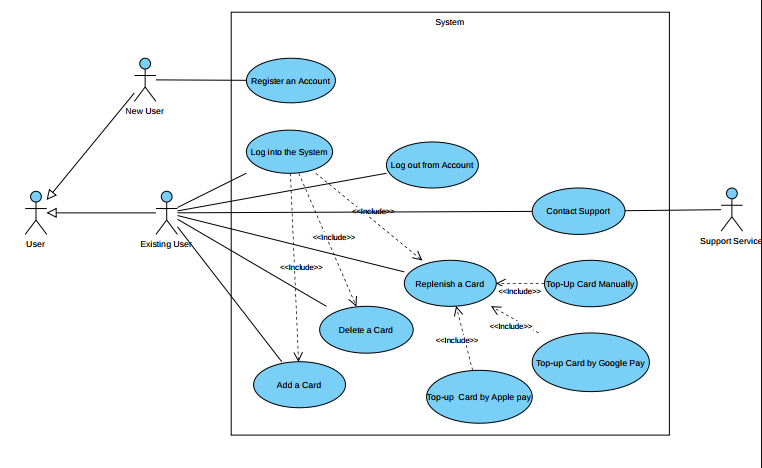
**User** – the only external actor. Since our system is fully automated there is no need in other external actors, such as operator or system analyst.

**Support Service**

## User Story Descriptions

|  |  |  |
| --- | --- | --- |
| User story name | Description | Release |
| Logging into the System | User sign into his account where he will be able to manage his cards. | R1 |
| User Registration | In case a user does not have an account, he will need to create a new account with a phone number which has not been used for other accounts before. | R2 |
| User Log out | User can log out from his own account. | R3 |
| Addition of Cards | To be able to manage cards a user should add cards using the ID of the card which is on the back side of it. | R4 |
| Balance Replenishment | This user story is the main function of our system. Using this function, a user will be able to replenish his card using the method of his choice. | R5 |
| Balance Replenishment Manually | One of the available methods to top-up cards is Manual. The user provides card details and System sends request to the debit card’s bank. | R6 |
| Balance Replenishment by Apple Pay | One of the available methods to top-up card is Apple Pay. The user that using application from IOS can make payment easier if Apple Pay already configured on his device | R7 |
| Balance Replenishment by Google Pay | One of the available methods to top-up card is Google Pay. The user that using application from Android OS can make payment easier if Google Pay already configured on his device | R8 |
| Card Deletion | A user should be able to delete his card from his account in case he loses it or gives it to someone else (Baku Card aren’t assigned to someone’s name and can be passed around). | R9 |
| Contact Support | A user should be able to contact a support service to resolve their problem. | R10 |
| Show Balance | User can check balance of Baku Card online 24/7 | R11 |
| Get Loan | User can get a loan once in a month for urgent situations | R12 |

### Use Case Diagram



## Use Case Descriptions

### Use Case 1

|  |  |
| --- | --- |
| **Use Case Number:** | UC-01 Priority: High |
| **Use Case Name:** | Log in the System |
| **Actor(s):** | Existing User |
| **Description:** | Existing User shall provide personal information to the System to be active User. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - Existing User logging into the System. |
| **Pre-condition(s):** | * System has been setup and configured. * System is running and open for login. * User granted internet access for the System |
| **Main (Success) Flow:** | 1. System requests login information  2. Existing User provides login information.  3. System verifies required information is provided.  ● If information is invalid System displays message. Return to Step 1  4. System displays main page to the User |
| **Alternate Flows:** | Alternate Flow #1: After Step 2 in success scenario System will display the option to Recover forgotten password. The following steps would occur:  1. Existing User selects option to Recover password during Login  2. System requests personal information to send a message with Recovery link   * If information is invalid System displays message. Return to Step 1 or exit * If information is invalid System provides option to Register an account if not exists.   3. System sends recovery link to user’s address for resetting password  4. Existing User opens link and provides new password for account  5. System verifies required information is provided  6. System returns to the login screen. |
| **Post Condition:** | System granted access to Existing User to main screen of System.  Loaded User’s information |
| **Requirements:** | 1 – The user shall login with personal information, 2 – The user shall be able to switch to registration during the process, 3- The system shall provide registration option if personal information given wrong. 4 – The system shall verify all information provided by user for registration as a patient. |

### Use Case 2

|  |  |
| --- | --- |
| **Use Case Number:** | UC-02 Priority: High |
| **Use Case Name:** | Register an Account |
| **Actor(s):** | New Customer |
| **Description:** | New Customer shall provide personal information to the System upon registering and becoming a User. |
| **Priority (Release)** | R2 |
| **Trigger:** | External - New Customer registering as a User. |
| **Pre-condition(s):** | * System has been setup and configured. * System is running and open for registrations. * User granted internet access for the System |
| **Main (Success) Flow:** | 1. New Customer selects option to register (phone number or email)  2. System requests personal information  3. Registrant provides personal information.  4. System verifies required information is provided.  ● If information is invalid System displays message. Return to Step 2  5. System requests login information  6. New Customer provides login information  7. System verifies required information is provided  ○ If information is invalid System displays message. Return to Step 5  8. System displays confirmation of registration |
| **Alternate Flows:** | Alternate Flow #1: After Step 2 in success scenario System will display the option to Cancel the registration process. The following steps would occur:  1. New Customer selects option to cancel during registration  2. System requests confirmation to cancel  3. New Customer confirms intent  4. System returns to main screen |
| **Post Condition:** | New Customer complete registration. System store Registrant's information. |
| **Requirements:** | 1 – The user shall register with personal information, 2 – The user shall be able to cancel registration during the process, 3– The system shall verify all information provided by user for registration as a User. |

### Use Case 3

|  |  |
| --- | --- |
| **Use Case Number:** | UC-03 Priority: High |
| **Use Case Name:** | Add a Card |
| **Actor(s):** | User |
| **Description:** | To replenish and manage his cards, a user needs to add his cards into the system and bind it to their account. |
| **Priority (Release)** | R4 |
| **Trigger:** | External – A user pressing “Add New Smart Card” |
| **Pre-condition(s):** | * A User must be logged into his account * The card he is trying to add can’t be assigned to another account. |
| **Main (Success) Flow:** | 1. User presses the “Add New Smart Card” button and is redirected to other page. 2. System asks for an ID code on the back of the card. 3. A user provides the ID. 4. If a card ID is correct and not taken by another user, the card is bound to the User’s account. 5. Systems redirects the User to the main page where their card can be seen and managed by them. |
| **Alternate Flows:** | Alternate Flow #1: In case a system says that the card number is incorrect, and you are sure that there is some problem in the system the following steps can be taken:   1. Contact Support service and inform them about the problem. 2. Wait for the response. |
| **Post Condition:** | A Baku CardBa is bound to User’s account and can be managed from main page. |
| **Requirements:** | 1 – The user shall provide his cards ID, 2 – The system shall verify all information provided by user for registration as a patient, 3 – The user shall be logged in into his account |

### Use Case 4

|  |  |
| --- | --- |
| **Use Case Number:** | UC-04 Priority: High |
| **Use Case Name:** | Replenish a Card |
| **Actor(s):** | User |
| **Description:** | This user story is the main function of our system. Using this function, a user will be able to replenish his card using the method of his choice. |
| **Priority (Release)** | R5 |
| **Trigger:** | External – A user pressing “Load Money” |
| **Pre-condition(s):** | * A User must be logged into his account. * At least one card should be bound to the account. |
| **Main (Success) Flow:** | 1. User presses the “Load Money” button and is redirected to another page. 2. User chooses the amount he wants to load to their balance. 3. Users choose the method with which they want to replenish their balance. 4. User verifies the transaction. 5. System adds money to the balance. |
| **Alternate Flows:** | Alternate Flow #1: After Step 2 in success scenario System will display the option to change the amount:   1. A user chooses to change the amount to load by pressing “Change”. 2. A user chooses a new amount to replenish. 3. A user proceeds with the main flow.   Alternate Flow #2: After Step 3 in success scenario System will display the option to change the method of payment:   1. A user chooses to change the amount to load by pressing “Change the method of payment”. 2. A user chooses a new method to replenish. 3. A user proceeds with the main flow.     Alternate Flow #1: During Step 4 in success scenario a user can choose to cancel the transaction completely:   1. A user chooses the “Cancel” option. 2. A user is redirected to the main page. |
| **Post Condition:** | The main goal is reached, the money is added to the balance. |
| **Requirements:** | 1 – The user shall verify the transaction his method of payment, 2 – The system shall verify all information provided by user for registration as a patient, 3 – The user shall be logged in into his account, 4 - The user shall have at least one card bound to their account. |

### Use Case 5

|  |  |
| --- | --- |
| **Use Case Number:** | UC-05 Priority: Low |
| **Use Case Name:** | Delete a Card |
| **Actor(s):** | User |
| **Description:** | A user should be able to delete his card from his account in case he loses it or gives it to someone else (Baku Card aren’t assigned to someone’s name and can be passed around). |
| **Priority (Release)** | R9 |
| **Trigger:** | External – A user pressing “Delete” a particular card. |
| **Pre-condition(s):** | * A User must be logged into his account * At least one card should be bound to the account. |
| **Main (Success) Flow:** | 1. A user presses “Delete” button for a particular card. 2. After pressing the “Delete”, a system asks if a user is sure about his decision. 3. In case user presses “yes” the card is unbound from user’s account. 4. A user is redirected to the main page. |
| **Alternate Flows:** | No Alternate Flows. |
| **Post Condition:** | A card is unbound from the user’s account and can’t be managed by them. |
| **Requirements:** | 1 – The user shall provide his card's ID, 2 – The system shall verify all information provided by user for registration as a patient, 3 – The user shall be logged in into his account, 4 - The user shall have at least one card bound to their account. |

### Use Case 6

|  |  |
| --- | --- |
| **Use Case Number:** | UC-06 Priority: High |
| **Use Case Name:** | Top-up Card Manually |
| **Actor(s):** | User |
| **Description:** | Since User already choice replenishment operation, one of the available methods to top-up card is Manual. The user provides card details to the System. |
| **Priority (Release)** | R6 |
| **Trigger:** | External – User pressing “Manual” replenishment |
| **Pre-condition(s):** | * User must be logged into his account * At least one card should be bound to the account. * User must choose replenishment operation and provide top-up amount (UC-04) * User must have positive balance on his bank card |
| **Main (Success) Flow:** | 1. System displays replenishment page to User 2. A User chooses Manual top-up option 3. System displays Manual top-up page to User 4. System requests debit card information 5. User provides debit card information  * If information is false, go back to Step 4 * If debit card balance is less than provided amount show error and back to the Step 4  1. System processes the transaction and updates User’s balance 2. System displays Success message and returns User to main page |
| **Alternate Flows:** | Alternate Flow #1: After Step 1 in success scenario System will display the option to cancel replenishment option. The following steps would be taken:  1. User selects option to cancel during replenishment  2. System requests confirmation to cancel  3. User confirms intent  4. System returns to main screen  Alternate Flow #2: After Step 5 in success scenario System will display the option to change debit card information. The following steps would be taken:  1. User selects option to change debit card information  2. System requests confirmation to change.  3. System continues operations from Success Flow Step 4 |
| **Post Condition:** | “Baku Card” balance topped-up by x amount of AZN. User’s debit card balance updated. |
| **Requirements:** | 1 – The user shall provide his debit card information, 2 – The system shall verify all information provided by user, 3 – The user shall be logged in into his account, 4 - The user shall have at least one card bound to their account. 5 – The user shall have a positive amount of money on the debit card balance. |

### Use Case 7

|  |  |
| --- | --- |
| **Use Case Number:** | UC-07 Priority: Medium |
| **Use Case Name:** | Top-up Card by Apple Pay |
| **Actor(s):** | User |
| **Description:** | Since User already choose replenishment operation, one of the available methods to top-up card is Apple Pay. The system asks Apple Pay for a transaction to replenish the balance. |
| **Priority (Release)** | R7 |
| **Trigger:** | External – User pressing “Apple Pay” replenishment |
| **Pre-condition(s):** | * User must be logged into his account * At least one card should be bound to the account. * User must choose replenishment operation and provide top-up amount (UC-04) * User must have positive balance on his bank card attached to Apple Pay * User must have IOS to use Apple Pay |
| **Main (Success) Flow:** | 1. System requests transaction from Apple Pay  2. Apple Pay sends request to User to confirm transaction  3. User confirms requests from Apple Pay  4. System confirms transaction from Apple Pay   * If the card balance attached to Apple Pay is less than the provided amount show error and back to Step 1  1. System processes the transaction and updates User’s balance 2. System displays Success message and returns User to main page |
| **Alternate Flows:** | Alternate Flow #1: After Step 1 in success scenario System will display the option to cancel replenishment option. The following steps would be taken:  1. User selects option to cancel during replenishment  2. System requests confirmation to cancel  3. User confirms intent  4. System returns to main screen |
| **Post Condition:** | “Baku Card” balance topped-up by x amount of AZN. User’s debit card balance updated. |
| **Requirements:** | 1 – The user shall provide his debit card information, 2 – The system shall verify all information provided by user, 3 – The user shall be logged in into his account, 4 - The user shall have at least one card bound to their account. 5 – The user shall have a positive amount of money on the debit card balance. |

### Use Case 8

|  |  |
| --- | --- |
| **Use Case Number:** | UC-08 Priority: Medium |
| **Use Case Name:** | Top-up Card by Google Pay |
| **Actor(s):** | User |
| **Description:** | Since User already choose replenishment operation, one of the available methods to top-up card is Google Pay. The system asks Google Pay for a transaction to replenish the balance. |
| **Priority (Release)** | R8 |
| **Trigger:** | External – User pressing “Google Pay” replenishment |
| **Pre-condition(s):** | * User must be logged into his account * At least one card should be bound to the account. * User must choose replenishment operation and provide top-up amount (UC-04) * User must have positive balance on his bank card attached to Google Pay * User must have Android OS to use Google Pay |
| **Main (Success) Flow:** | 1. System requests transaction from Google Pay  2. Apple Pay sends request to User to confirm transaction  3. User confirms requests from Google Pay  4. System confirms transaction from Google Pay   * If the card balance attached to Google Pay is less than the amount provided show error and back to Step 1  1. System processes the transaction and updates User’s balance 2. System displays Success message and returns User to main page |
| **Alternate Flows:** | Alternate Flow #1: After Step 1 in success scenario System will display the option to cancel replenishment option. The following steps would be taken:  1. User selects option to cancel during replenishment  2. System requests confirmation to cancel  3. User confirms intent  4. System returns to main screen |
| **Post Condition:** | “Baku Card” balance topped-up by x amount of AZN. User’s debit card balance updated. |
| **Requirements:** | 1 – The user shall provide his debit card information, 2 – The system shall verify all information provided by user, 3 – The user shall be logged in into his account, 4 - The user shall have at least one card bound to their account. 5 – The user shall have a positive amount of money on the debit card balance. |

### Use Case 9

|  |  |
| --- | --- |
| **Use Case Number:** | UC-09 Priority: High |
| **Use Case Name:** | Log out from Account |
| **Actor(s):** | User |
| **Description:** | In case a user needs to enter another account, he should be able to log out from the account to which he is signed in currently. |
| **Priority (Release)** | R3 |
| **Trigger:** | External – A user pressing “Log Out” button. |
| **Pre-condition(s):** | * A User must be logged into his account |
| **Main (Success) Flow:** | 1. A user presses “Log Out” button. 2. The system logs out a user from his account. 3. System redirects user to log in page. |
| **Alternate Flows:** | Alternate Flow #1: In case a connection is lost a user should follow these steps:   1. A user should ensure that he has a stable connection. 2. Proceed with Main Flow. |
| **Post Condition:** | A user is logged out of his account and redirected to log in page. |
| **Requirements:** | 1– The user shall be logged into his account, 2 - The user shall have stable Internet connection. |

### Use Case 10

|  |  |
| --- | --- |
| **Use Case Number:** | UC-10 Priority: Low |
| **Use Case Name:** | Contact Support |
| **Actor(s):** | User, Support Service |
| **Description:** | A user should be able to contact a support service to resolve their problem. |
| **Priority (Release)** | R10 |
| **Trigger:** | External – A user pressing “Log Out” button. |
| **Pre-condition(s):** | * No preconditions needed |
| **Main (Success) Flow:** | 1. A user contacts support service through an app feature. 2. A support service answers a customer. 3. A support service operator tries to do everything possible to solve the problem |
| **Alternate Flows:** | Alternate Flow #1: In case a support answer does not answer through app a user should proceed with following steps:   1. A customer calls support service hotline. 2. Step 2 of Main Flows are followed. |
| **Post Condition:** | A user is logged out of his account and redirected to log in page. |
| **Requirements:** | 1– The user shall be logged into his account, 2 - The user shall have stable Internet connection. |

### Use Case 11

|  |  |
| --- | --- |
| **Use Case Number:** | UC-11 Priority: Medium |
| **Use Case Name:** | Show Balance |
| **Actor(s):** | User |
| **Description:** | User checks Baku Card’s balance online |
| **Priority (Release)** | R11 |
| **Trigger:** | External – A user checks card balance |
| **Pre-condition(s):** | * User must be logged into his account * User must have attached card to his account |
| **Main (Success) Flow:** | 1. System checks attached Baku Card’s balance from database 2. System displays updated Baku Card’s balance |
| **Alternate Flows:** | No Alternate Flows |
| **Post Condition:** | Updated balance is shown to user |
| **Requirements:** | 1– The user shall be logged into his account, 2 - The user shall have stable Internet connection. 3 – The user shall have attached Baku Card to its account. |

### Use Case 12

|  |  |
| --- | --- |
| **Use Case Number:** | UC-12 Priority: Low |
| **Use Case Name:** | Get Loan |
| **Actor(s):** | User |
| **Description:** | Each user can have a once in a month loan of 0.3 AZN |
| **Priority (Release)** | R12 |
| **Trigger:** | External – A user gets loan on its Baku Card |
| **Pre-condition(s):** | * User must be logged into his account * User must have attached card to his account * The time from the date of purchase of the Baku Card must exceed 30 days. |
| **Main (Success) Flow:** | 1. User presses the “Loan” button 2. System checks request sent by user to get the loan  * If the previous loan is not repaid, display the appropriate error message and cancel the request. * If 30 days have not passed since the previous loan was taken, show the appropriate error message and cancel the request.  1. System grants user spend Baku Card balance up to -0.3 AZN once during 24 hours from the moment of processing a loan request. 2. The system prohibits user from spending Baku Card balance in Loan mode if card balance is seen below zero and resets limitations spending limitations till 0.0 AZN |
| **Alternate Flows:** | Alternate Flow #1: In case a user didn’t spend his Loan option (balance isn’t seen below zero) system should proceed with following steps:   1. System resets user’s loan options after 24hours of not using and gives chance to take it anytime user wants. 2. The system updates the countdown timer as requested (resets). |
| **Post Condition:** | For subsequent use of the card, the user must first repay the loan. |
| **Requirements:** | 1– The user shall be logged into his account, 2 - The user shall have stable Internet connection. 3 – The user shall have attached Baku Card to its account. |

# References

1)<https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>