



**Revision and Sign Off Sheet**

**Change Record**

| **Author** | **Version** | **Change reference** | **Date** |
| --- | --- | --- | --- |
| Quynh Ton | 0.5.0 | Complete requirement direction | 25/11/2021 |
| Quynh Ton | 0.7.0 | Complete requirement direction | 04/12/2021 |
| Quynh Ton | 1.0.0 | The final | 08/12/2021 |

**Reviewers**

| **Name** | **Company** | **Version** | **Position** | **Date** |
| --- | --- | --- | --- | --- |
| Trinh-Dong Nguyen | FPT | 0.5.0 | AD Team Lead |  |
|  |  |  |  |  |

**Table of Contents**

[1.](#_heading=h.gjdgxs) Introduction 6

[1.1.](#_heading=h.30j0zll) Purpose 6

[1.2.](#_heading=h.1fob9te) Scope 6

[1.3.](#_heading=h.3znysh7) Intended Audiences and Document Organization 6

[1.4.](#_heading=h.3dy6vkm) References 7

[2.](#_heading=h.4d34og8) Functional Requirements 8

[2.1.](#_heading=h.2s8eyo1) Use Case Description 8

[2.1.1.](#_heading=h.17dp8vu) Sign up 8

[2.1.2.](#_heading=h.lnxbz9) Sign in 12

[2.1.3.](#_heading=h.44sinio) Sign out 15

[2.1.4.](#_heading=h.3j2qqm3) Remember account 18

[2.1.5.](#_heading=h.1ci93xb) Forgot password 21

[2.1.6.](#_heading=h.qsh70q) Change password 26

[2.1.7.](#_heading=h.49x2ik5) Link bank account 28

[2.1.8.](#_heading=h.3o7alnk) Deposit money into digital wallet 32

[2.1.9.](#_heading=h.32hioqz) Upgrade account 34

[2.1.10.](#_heading=h.2grqrue) Get notification 37

[2.1.11.](#_heading=h.1v1yuxt) Filter users 40

[2.1.12.](#_heading=h.3tbugp1) View general user information 44

[2.1.13.](#_heading=h.37m2jsg) Send matching request 46

[2.1.14.](#_heading=h.46r0co2) Receive matching request 48

[2.1.15.](#_heading=h.3l18frh) Create chat room 51

[2.1.16.](#_heading=h.2zbgiuw) View chatting history 54

[2.1.17.](#_heading=h.2dlolyb) Chat 57

[2.1.18.](#_heading=h.1rvwp1q) Edit personal information 61

[2.1.19.](#_heading=h.1664s55) Review app 64

[2.1.20.](#_heading=h.kgcv8k) Report user 66

[2.1.21.](#_heading=h.43ky6rz) Report incident 69

[2.1.22.](#_heading=h.3hv69ve) Create video call room 72

[2.1.23.](#_heading=h.2w5ecyt) View calling history 75

[2.1.24.](#_heading=h.2afmg28) View matching history 77

[2.1.25.](#_heading=h.1opuj5n) Search user 80

[2.1.26.](#_heading=h.1302m92) Match with searching user 83

[2.1.27.](#_heading=h.haapch) CRUD Status 84

[2.1.28.](#_heading=h.1gf8i83) React to user post 87

[2.1.29.](#_heading=h.2fk6b3p) Add to the Favourites list 88

[2.1.30.](#_heading=h.1tuee74) View favourites list 90

[2.1.31.](#_heading=h.184mhaj) Block user 92

[2.1.32.](#_heading=h.meukdy) Unblock user 94

[2.1.33.](#_heading=h.45jfvxd) Boost account 96

[2.1.34.](#_heading=h.3jtnz0s) Create user statistics table 97

[2.1.35.](#_heading=h.2y3w247) View user information 99

[2.1.36.](#_heading=h.2ce457m) View users’ posts 102

[2.1.37.](#_heading=h.1qoc8b1) View user reports 104

[2.1.38.](#_heading=h.3oy7u29) Lock account 106

[2.1.39.](#_heading=h.338fx5o) Send warning to user 109

[2.2.](#_heading=h.2hio093) UI Description 111

[2.3.](#_heading=h.3gnlt4p) Database Description 111

[3.](#_heading=h.1vsw3ci) Non-functional Requirements 111

[3.1.](#_heading=h.4fsjm0b) User Access and Security 111

[3.2.](#_heading=h.2981zbj) Performance Requirements 113

[3.3.](#_heading=h.odc9jc) Implementation Requirements 113

[4.](#_heading=h.38czs75) Other Requirements 114

[4.1.](#_heading=h.1nia2ey) Archive Function 114

[4.2.](#_heading=h.47hxl2r) Security Audit Function 114

[5.](#_heading=h.2mn7vak) Các yêu cầu hệ thống 114

[Sites 114](#_heading=h.11si5id)

[Lists 114](#_heading=h.3ls5o66)

[Permission Levels 114](#_heading=h.302dr9l)

[Groups 115](#_heading=h.1f7o1he)

[Web Parts 115](#_heading=h.3z7bk57)

[Custom Pages 116](#_heading=h.2eclud0)

[Scheduled Agents 116](#_heading=h.thw4kt)

[Technical Concern 116](#_heading=h.3dhjn8m)

[6.](#_heading=h.1smtxgf) Appendixes 116

[6.1.](#_heading=h.4cmhg48) Glossary 116

[6.2.](#_heading=h.2rrrqc1) Mapping to Notes Application 117

[6.3.](#_heading=h.16x20ju) Messages 117

[6.4.](#_heading=h.356xmb2) Issues List 118

**LIST OF FIGURES**

[Figure 1: Activity Flow - Sign up 9](#_heading=h.3rdcrjn)

[Figure 2: Sequence Diagram - Sign up 10](#_heading=h.26in1rg)

[Figure 3: Activity Flow - Sign in 13](#_heading=h.35nkun2)

[Figure 4: Sequence Diagram - Sign in 14](#_heading=h.1ksv4uv)

[Figure 5: Activity Flow - Sign out 16](#_heading=h.2jxsxqh)

[Figure 6: Sequence Diagram - Sign out 17](#_heading=h.z337ya)

[Figure 7: Activity Flow - Remember account (1) 19](#_heading=h.1y810tw)

[Figure 8: Activity Flow - Remember Account (2) 19](#_heading=h.4i7ojhp)

[Figure 9: Sequence Diagram - Remember account 20](#_heading=h.2xcytpi)

[Figure 10: Activity Flow - Forgot password 22](#_heading=h.3whwml4)

[Figure 11: Sequence Diagram - Forgot password 23](#_heading=h.2bn6wsx)

[Figure 12: Activity Flow - Change password 26](#_heading=h.3as4poj)

[Figure 13: Sequence Diagram - Change password 27](#_heading=h.1pxezwc)

[Figure 14: Activity Flow - Link bank account 29](#_heading=h.2p2csry)

[Figure 15: Sequence Diagram - Link bank account 30](#_heading=h.147n2zr)

[Figure 16: Activity Flow - Deposit money into digital wallet 33](#_heading=h.23ckvvd)

[Figure 17: Sequence Diagram - Deposit money into digital wallet 33](#_heading=h.ihv636)

[Figure 18: Activity Flow - Upgrade account 35](#_heading=h.1hmsyys)

[Figure 19: Sequence Diagram - Upgrade account 36](#_heading=h.41mghml)

[Figure 20: Activity Flow - Get notification 38](#_heading=h.vx1227)

[Figure 21: Sequence Diagram - Get notification 38](#_heading=h.3fwokq0)

[Figure 22: Activity Flow - Filter users 41](#_heading=h.4f1mdlm)

[Figure 23: Sequence Diagram - Filter users 41](#_heading=h.2u6wntf)

[Figure 24: Activity Flow - View general user information 44](#_heading=h.28h4qwu)

[Figure 25: Sequence Diagram - View general user information 45](#_heading=h.nmf14n)

[Figure 26: Activity Flow - Send matching request 47](#_heading=h.1mrcu09)

[Figure 27: Activity Flow - Receive matching request 49](#_heading=h.2lwamvv)

[Figure 28: Sequence Diagram - Send matching request 50](#_heading=h.111kx3o)

[Figure 29: Activity Flow - Create chat room 52](#_heading=h.206ipza)

[Figure 30: Sequence Diagram - Create chat room 53](#_heading=h.4k668n3)

[Figure 31: Activity Flow - View chatting history 55](#_heading=h.1egqt2p)

[Figure 32: Sequence Diagram - View chatting history 56](#_heading=h.3ygebqi)

[Figure 33: Activity Flow - Chat 58](#_heading=h.sqyw64)

[Figure 34: Sequence Diagram – Chat 59](#_heading=h.3cqmetx)

[Figure 35: Activity Flow - Edit personal information 62](#_heading=h.4bvk7pj)

[Figure 36: Sequence Diagram - Edit personal information 62](#_heading=h.2r0uhxc)

[Figure 37: Activity Flow - Review app 65](#_heading=h.3q5sasy)

[Figure 38: Sequence Diagram - Review app 65](#_heading=h.25b2l0r)

[Figure 39: Activity Flow - Report user 67](#_heading=h.34g0dwd)

[Figure 40: Sequence Diagram - Report user 68](#_heading=h.1jlao46)

[Figure 41: Activity Flow - Report incident 70](#_heading=h.2iq8gzs)

[Figure 42: Sequence Diagram - Report incident 71](#_heading=h.xvir7l)

[Figure 43: Activity Flow - Create video call room 73](#_heading=h.1x0gk37)

[Figure 44: Sequence Diagram - Create video call room 74](#_heading=h.4h042r0)

[Figure 45: Activity Flow - View calling history 76](#_heading=h.1baon6m)

[Figure 46: Sequence Diagram - View calling history 76](#_heading=h.3vac5uf)

[Figure 47: Activity Flow - View matching history 78](#_heading=h.pkwqa1)

[Figure 48: Sequence Diagram - View matching history 79](#_heading=h.39kk8xu)

[Figure 49: Activity Flow - Search user 81](#_heading=h.48pi1tg)

[Figure 50: Sequence Diagram - Search user 82](#_heading=h.2nusc19)

[Figure 51: Activity Flow - Match with searching user 83](#_heading=h.3mzq4wv)

[Figure 52: Sequence Diagram - Match with searching user 84](#_heading=h.2250f4o)

[Figure 53: Activity Flow - CRUD Status 86](#_heading=h.319y80a)

[Figure 54: Activity Flow - React to user post 88](#_heading=h.40ew0vw)

[Figure 55: Activity Flow - Add to the favourites list 89](#_heading=h.upglbi)

[Figure 56: Sequence Diagram - Add to the favourites list 89](#_heading=h.3ep43zb)

[Figure 57: Activity Flow - View Favourites list 90](#_heading=h.4du1wux)

[Figure 58: Sequence Diagram - View Favourites list 91](#_heading=h.2szc72q)

[Figure 59: Activity Flow - Block user 93](#_heading=h.3s49zyc)

[Figure 60: Sequence Diagram - Block user 93](#_heading=h.279ka65)

[Figure 61: Activity Flow - Unblock user 94](#_heading=h.36ei31r)

[Figure 62: Sequence Diagram - Unblock user 95](#_heading=h.1ljsd9k)

[Figure 63: Activity Flow - Boost account 96](#_heading=h.2koq656)

[Figure 64: Sequence Diagram - Boost account 97](#_heading=h.zu0gcz)

[Figure 65: Activity Flow - Create user statistics table 98](#_heading=h.1yyy98l)

[Figure 66: Sequence Diagram - Create user statistics table 98](#_heading=h.4iylrwe)

[Figure 67: Activity Flow - View user information 100](#_heading=h.1d96cc0)

[Figure 68: Sequence Diagram - View user information 101](#_heading=h.3x8tuzt)

[Figure 69: Activity Flow - View user posts 103](#_heading=h.rjefff)

[Figure 70: Sequence Diagram - View user posts 103](#_heading=h.3bj1y38)

[Figure 71: Activity Flow - Approve post 105](#_heading=h.4anzqyu)

[Figure 72: Activity Flow - View user reports 106](#_heading=h.2pta16n)

[Figure 73: Sequence Diagram - View user reports 106](#_heading=h.14ykbeg)

[Figure 74: Activity Flow - Lock account 108](#_heading=h.243i4a2)

[Figure 75: Sequence Diagram - Lock account 109](#_heading=h.j8sehv)

[Figure 76: Activity Flow - Send warning to user 110](#_heading=h.1idq7dh)

[Figure 77: Sequence Diagram - Send warning to user 111](#_heading=h.42ddq1a)

# Introduction

# Purpose

This Software Requirements Specification and Design document contains the software requirements to create Social Networking Dating application. Firstly, this document along with other reference documents are complete requirements. Secondly, it defines, technically, how applications will operate. Developers will base on this document to conduct development plan, task assignment and implementation of the new application.

# Scope

This document is prepared for the application, in scope of the project. The details include:

Time: the application will be completed in 6 months

Cost: the total money invest in this project is $22.000

Goal: building the application to generate connections between people who are interested in romance, casual sex, or friendship. Based on factors like age, gender of user and desired partner, the hobbies and the distance users will travel to meet one another, the application will recommend the compatible partner for user. In addition, the application also builds the digital wallet which helps users to deposit money and use this amount to unlock advanced features.

# Intended Audiences and Document Organization

This document is intended for:

* Development team: Responsible to develop detailed design, implement and perform unit test, integration test and system test for the migrated application
* Data Migration team: Responsible for creating data migration scripts and performing data migration for the application.
* Documentation Team: Responsible for writing User Guide for the application.
* UAT team: Responsible for conducting user acceptance test sessions with end users.

Below are main sections of the document:

* **1. Introduction**: This section describes the general introduction of this document.
* **2. Functional Requirements**: This section describesthe functional requirements in detail.
* **3. Non-functional Requirements:** This section describes the non-functional requirements of this application such as user access and security, interfaces, screens, and performance.
* **4. Other Requirements:** This section describes other requirements such as archive or security audit function.
* **5.** **Application Design:** This section describes the design of SharePoint application.
* **6. Appendixes**: This section describes other requirements for this application and other supporting information for this document**.**
* NOTE:Please refer to section 6.1 for all acronyms and abbreviations you may encounter within this document.

# References

| **#** | **Title** | **Version** | **File Name / Link** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | **Matching and Sorting in Online Dating** |  |  | Article in American Economic Review (March 2010). This paper studies the economics of match formation using a novel dataset obtained from a  major online dating service. |
| 2 | **Mate Preferences and Matching** |  |  | Article belongs to University of Chicago (April 2006). This paper uses a novel data set obtained from an online dating service to draw inferences on mate preferences and to investigate the role played by these preferences in determining match outcomes and sorting patterns. |
| 3 | **The Role of Physical Attractiveness in Online Dating Self-Presentation and Deception** |  |  | Article in Communication Research (May 2010). This study examines the role of online daters’ physical attractiveness in their profile self-presentation and their use of deception. |
| 4 | **Interaction-Based Collaborative Filtering Method for Recommendation in Online Dating** |  |  | A research paper of School of Computer Science and Engineering,  University of New South Wales  (Sydney, Australia). |
| 5 | **Recommendation System for Online Dating** |  |  | A research paper of the Department of Software Engineering, Charles University (Czech Republic). |

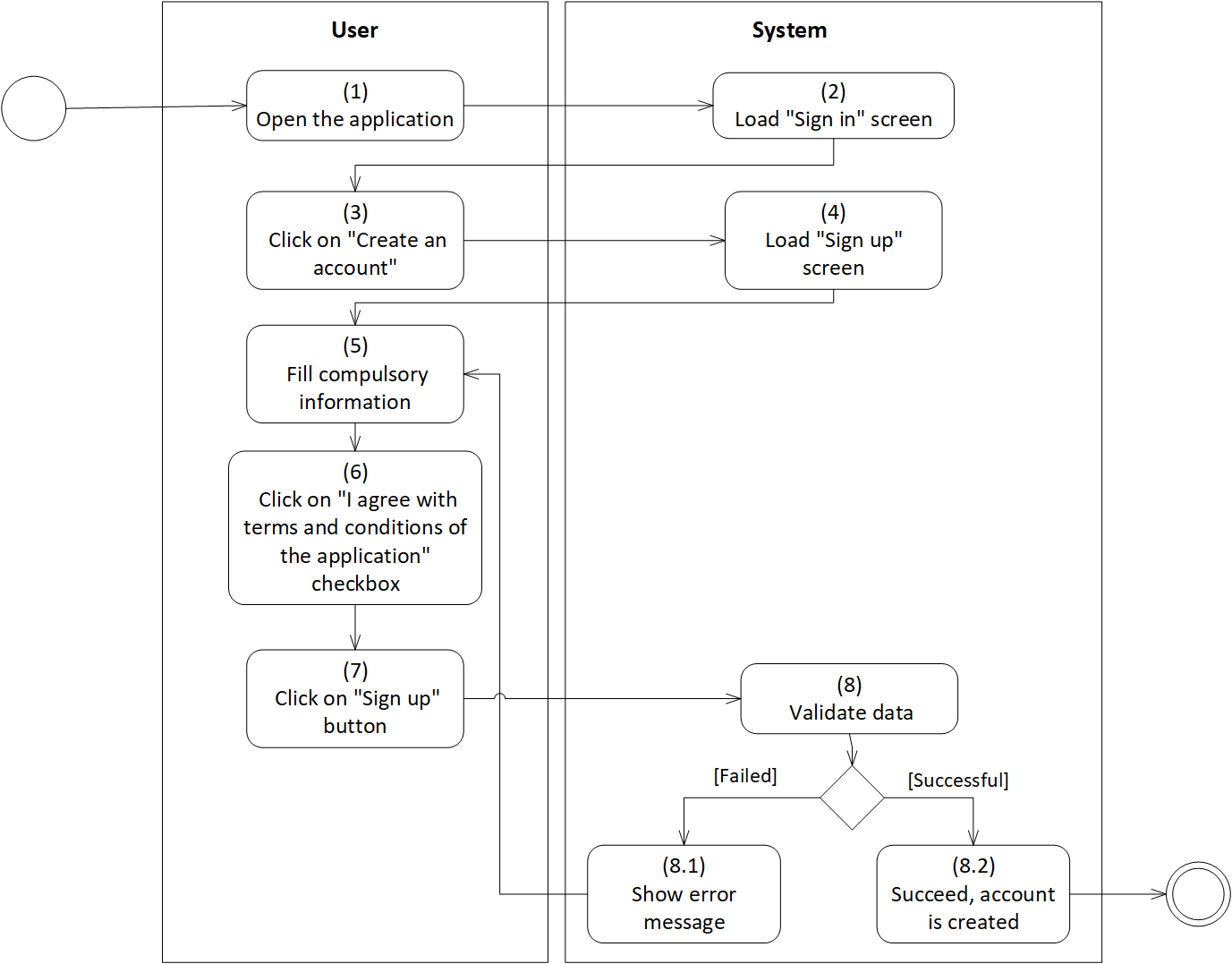
# Functional Requirements

# Use Case Description

# Sign up

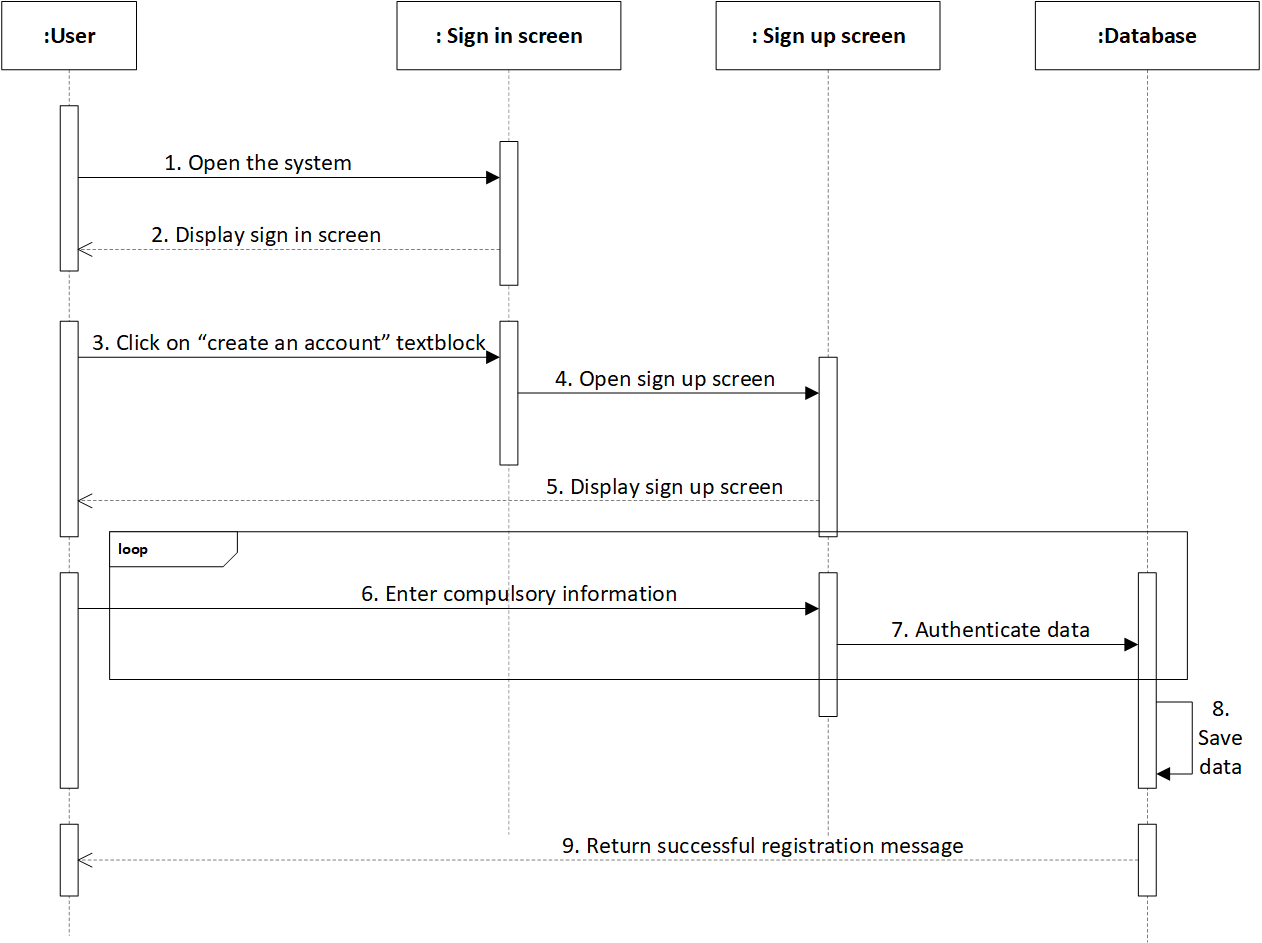
| **Name** | **Sign up** |
| --- | --- |
| **Description** | This use case allows user to create an account to access the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * User wants to access the Social Networking Dating application * Open the application * Click on “Sign up” button |
| **Pre-condition** | * Must be connected to the network |
| **Post-condition** | * User account is created * User can use this account to sign in. |

**Activities Flow**



*Figure 1: Activity Flow - Sign up*

**Sequence Diagram**



*Figure 2: Sequence Diagram - Sign up*

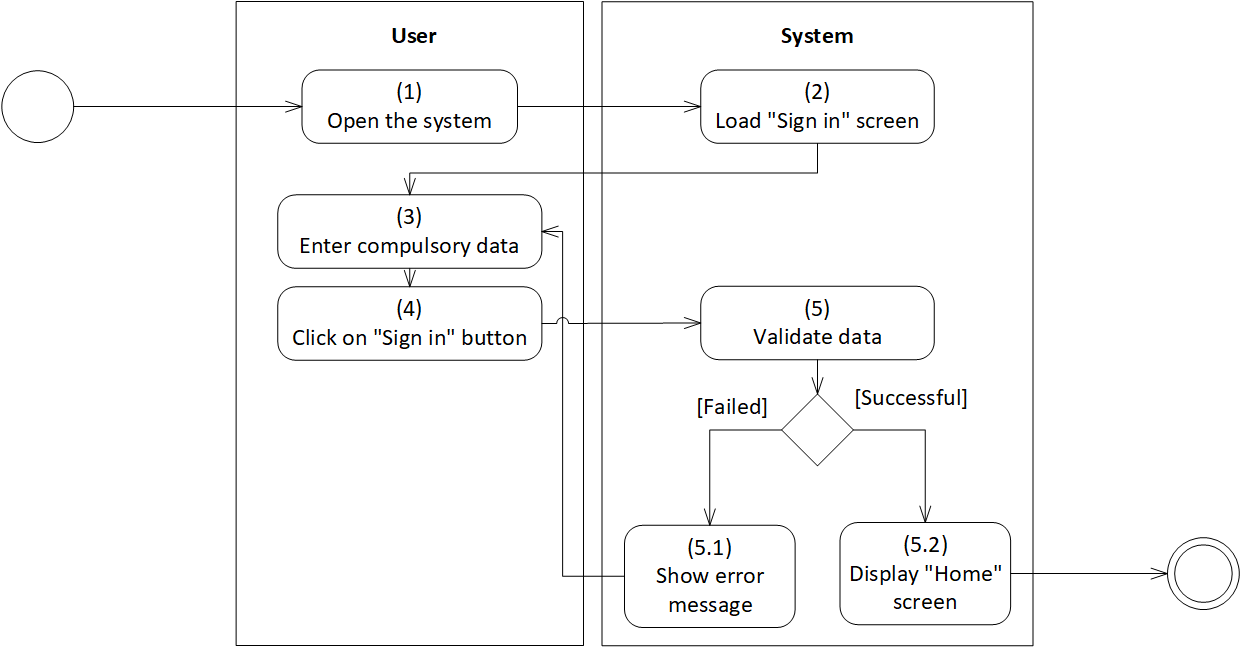
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Sign in” Form rules**:   * The system loads “Sign in” form (Refer to “Sign in” list in “UI Description” file). |
| *(4)* | *BR2* | **Load “Sign up” Form rules:**   * When user clicks on “Create an account”, the system will load “Sign up” form (Refer to “Sign up” list in “UI Description” file) |
| *(5), (6)* | *BR3* | **Input checking rules:**  User enters compulsory information in the textboxes of “Sign up” form. If user enters according to the accurate format, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If data textboxes do not follow the input rules, set [Textbox Colour] = “Red” * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else If type of [Textbox Text] != [Specified Data Type], set [Text Error] = “Please enter”+“<<Field Name>> + “as”+ <<Specified Data Type>> * Else If [Textbox Text] do not contains all types of character, set [Text Error] = “<<Field Name>>” + “must have” + “<<List of character types>>” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” * Else if format of [Textbox Text] is inaccurate, set [Text Error] = “Please enter correct format” * If data textboxes are inputted exactly, set [Textbox Colour] == “Green” |
|  |  | **Sign up Form**:  Fill information in “Sign up” form as the template below:   | Full name | * Textbox * Length [Full Name] >= 8 | | --- | --- | | Date of birth | * Date Time Picker * [Date Of Birth] = “dd-mm-YYYY” | | Phone No | * Textbox * Length [Phone No] >= 10 & [Phone No] <=11 | | Email | * Textbox * Check valid email by using a regular expression: @"^[^@\s]+@[^@\s]+\.[^@\s]+$" | | Address | * Combo box includes name of provinces * If [Address] != [Name] in [List of provinces], user can manually enter. | | Username | * Textbox * Length [Username] >= 8 * [Username] contains uppercase, lowercase, number, and special character. | | Password | * Textbox * Length [Password] >= 6 | | Confirmed password | * Textbox * Check validation: [Confirm Password] = [Password] | | Terms & Conditions | * Checkbox * User clicks on “Terms & Conditions” button, load “T&C” screen. * If [isChecked] = False, user cannot create an account. Else, continue validating other data. | |
| *(8)* | *BR4* | **Validate data rules:**   * If the data is processed successfully, the system will prompt a message (Refer to MSG3) * Else, the system will prompt an error message (Refer to MSG0) |

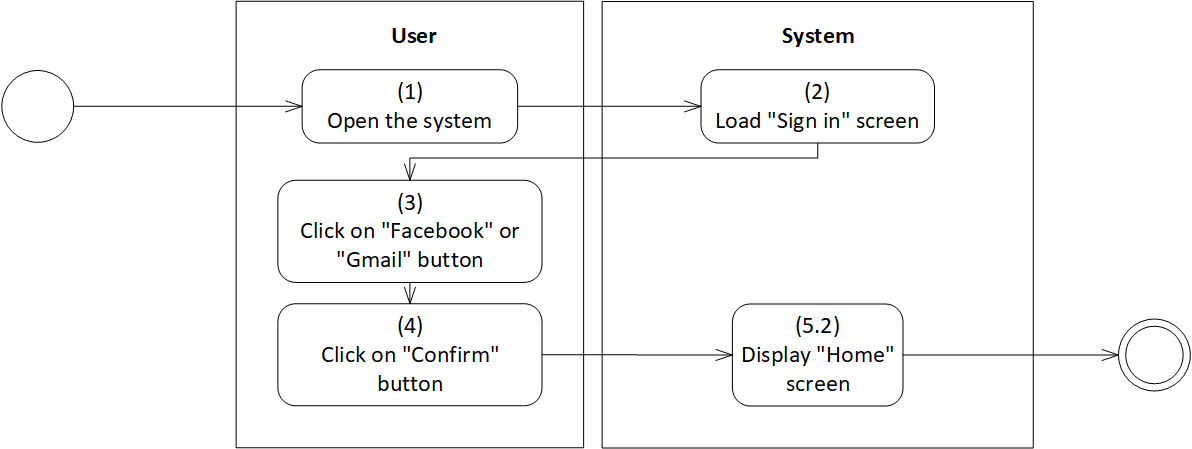
# Sign in

| **Name** | **Sign in** |
| --- | --- |
| **Description** | This use case allows user to be logged into the system |
| **Actor** | Normal User, Silver User, Gold User, Administrator |
| **Trigger** | * User wants to access the Social Networking Dating application * Open the application * Click on “Sign in” button * Or click on “Facebook” or “Gmail” button |
| **Pre-condition** | * Must be connected to the network * [isLogin]==False * Has account already |
| **Post-condition** | * User can access the home screen |

**Activity Flow**

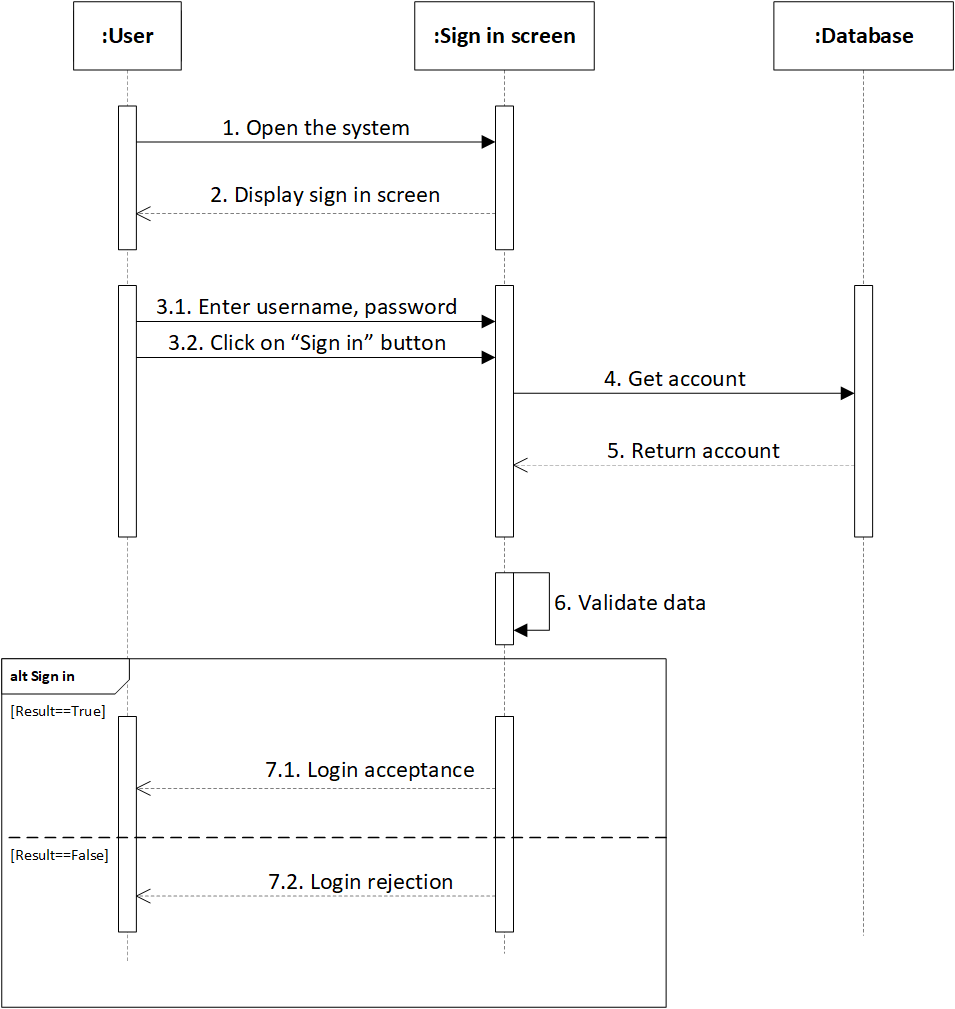


*Figure 3: Activity Flow - Sign in (1)*



*Figure 4: Activity Flow - Sign in (2)*

**Sequence Diagram**



*Figure 5: Sequence Diagram - Sign in*

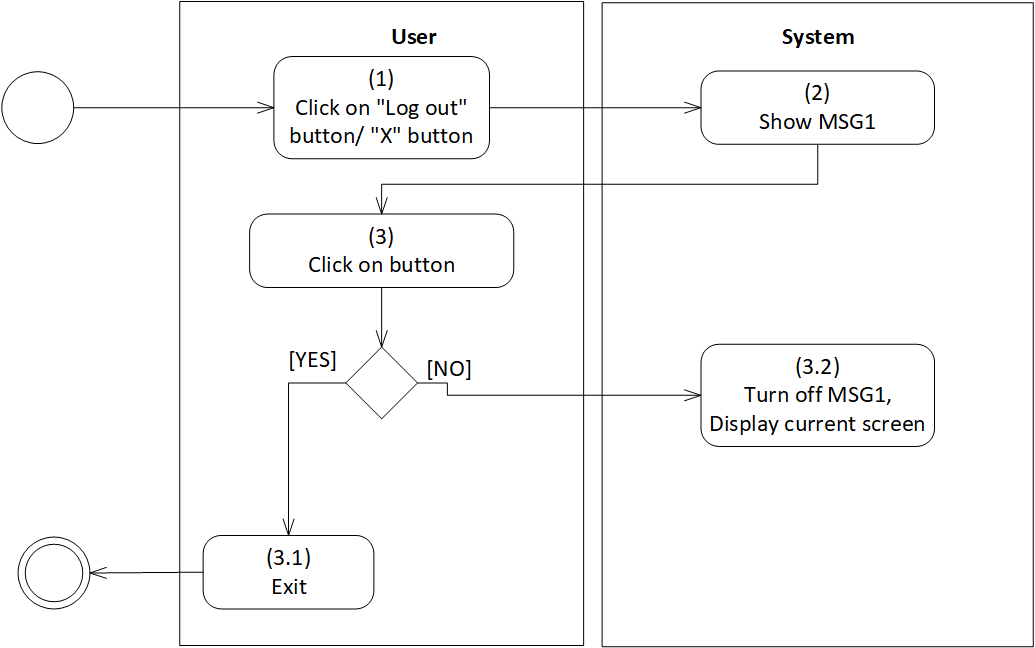
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Sign in” Form rules**:  The system loads “Sign in” form. (Refer to “Sign in” list in “UI Description” file) |
| *(3)* | *BR2* | **Input checking rules:**  User enters username and password in the textboxes of “Sign in” form. If user fills in full information, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Textbox Colour] = “Red” and set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” |
|  |  | **Sign in Form**:  Fill information in “Sign in” form as the template below:   | Username | Input [Username] of current user | | --- | --- | | Password | Input [Password] of current user | |
| *(4), (5)* | *BR3* | **Validating rules:**  When user clicks on “Sign in” button, the system will check the internet connection.   * If [CheckInternetConnect] = False, set [Text Error] = “No internet connection. Please connect to the internet right now”. * Else, the system will validate data as the following: * The system retrieves [Username] and [Password] in the database and checks whether [Username] in the textbox = [Username], [Password] in the textbox = [Password]. If yes, the system will prompt a message that notifies of successfully signing up (Refer to MSG3). * If not, the system will prompt an error message (Refer to MSG0). |

# Sign out

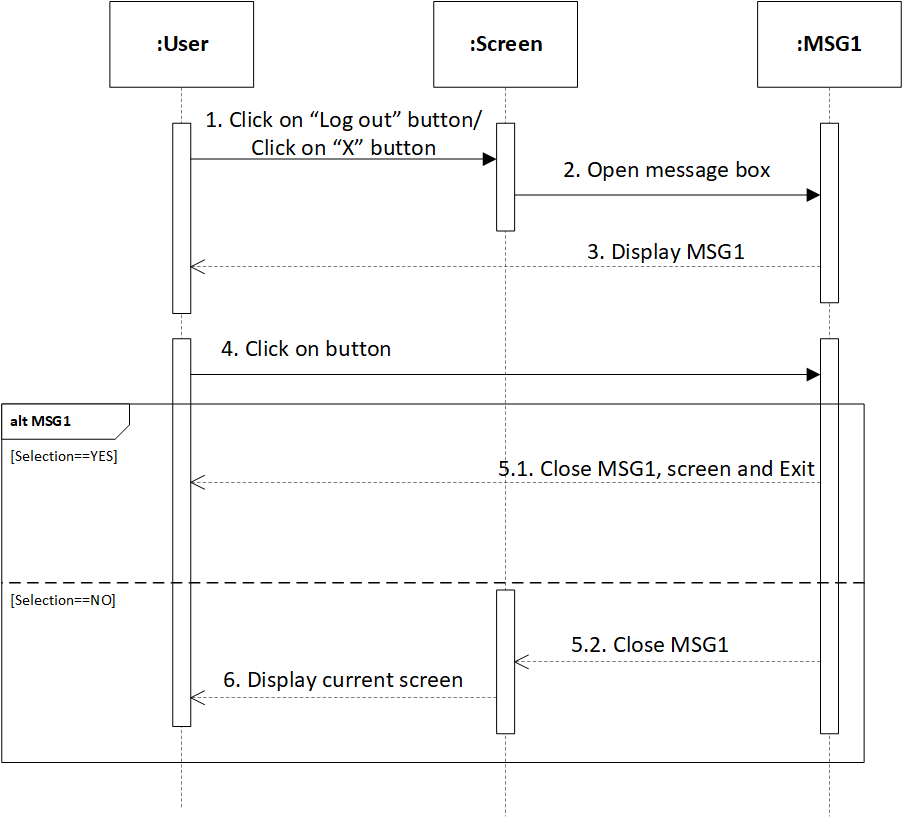
| **Name** | **Sign out** |
| --- | --- |
| **Description** | This use case allows user to be logged out the system |
| **Actor** | Normal User, Silver User, Gold User, Administrator |
| **Trigger** | * User wants to exit from the Social Networking Dating application |
| **Pre-condition** | * Must be connected to the network * Has signed in already or [isLogin]==True |
| **Post-condition** | * User can not have any manipulation on the system |

**Activity Flow**



*Figure 6: Activity Flow - Sign out*

**Sequence Diagram**



*Figure 7: Sequence Diagram - Sign out*

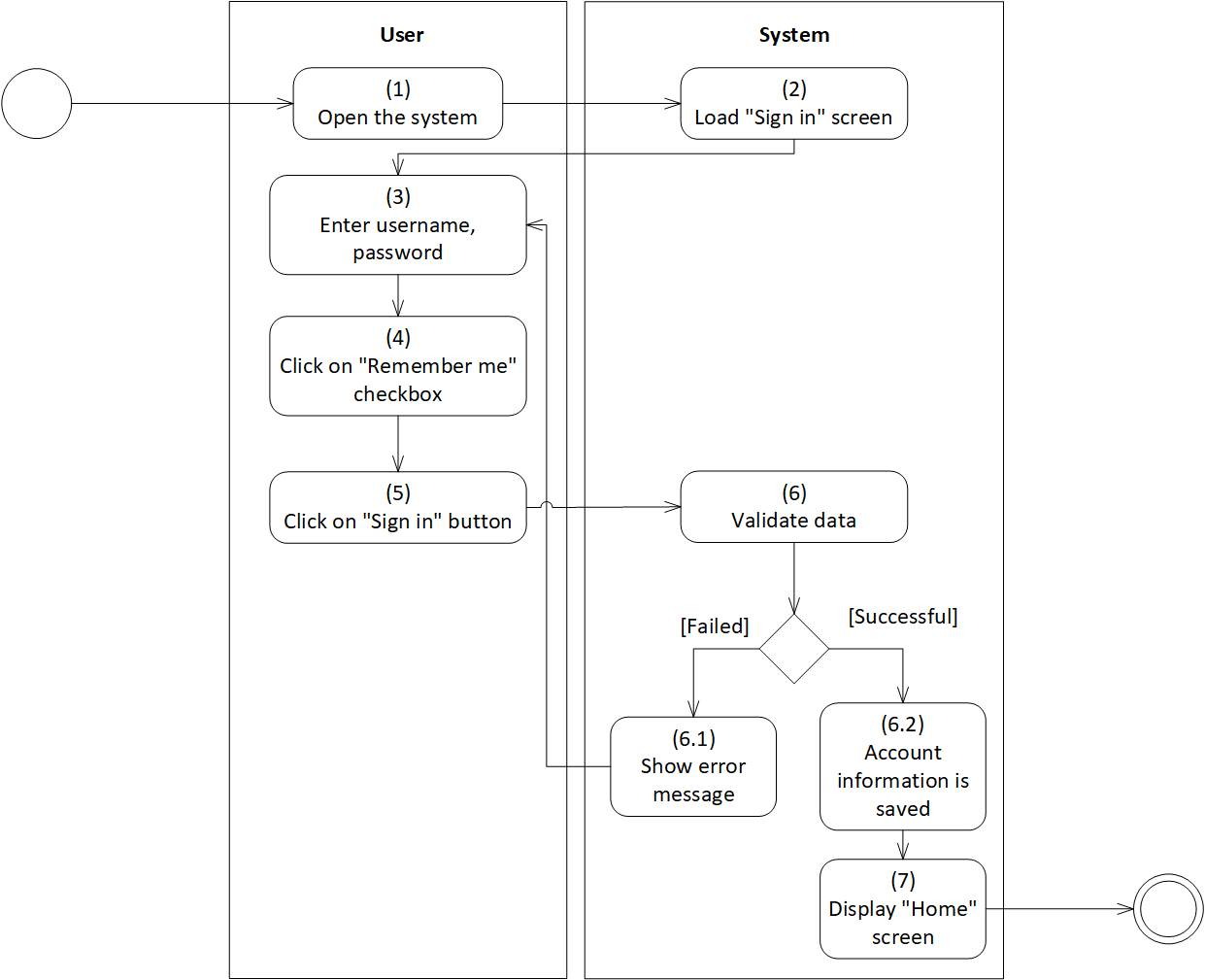
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load MSG1 rules**:  When user clicks on “Sign out” button on the menu sidebar or click on “X” button at the top right corner, the system will load a message. |
| *(3)* | *BR2* | **Selecting rules:**  When user clicks on one of two buttons on MSG1, the system will response as the following:   * If [Selection] = YES, close MSG1 and close the application. * Else, close MSG1 and turn back the current screen. |

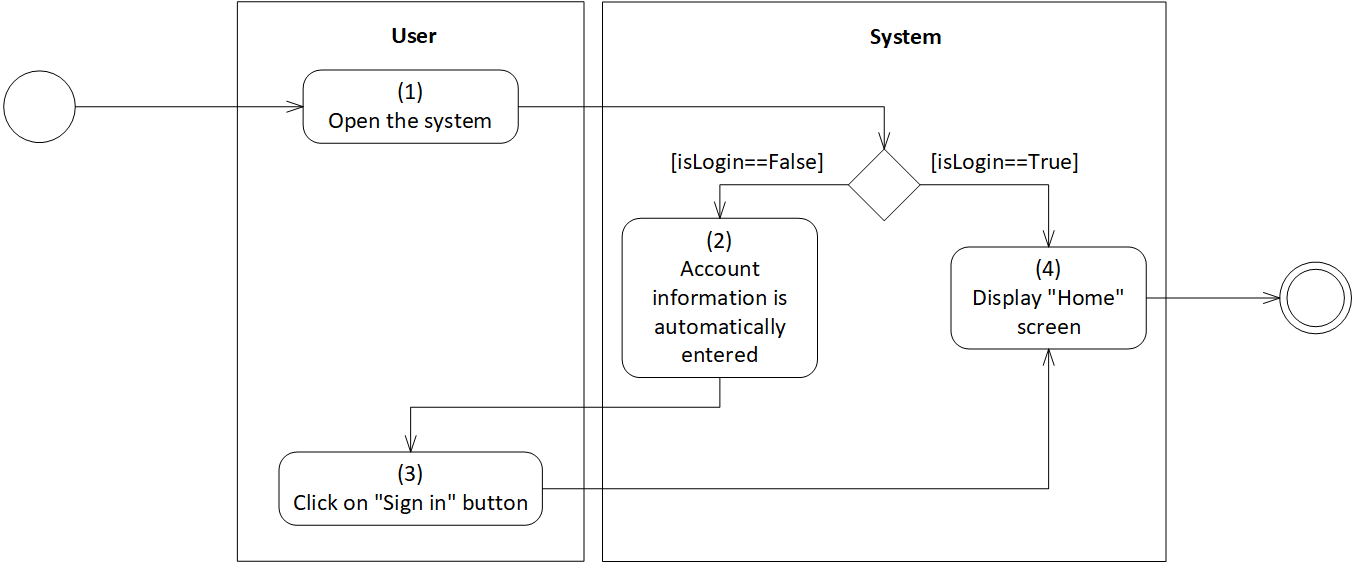
# Remember account

| **Name** | **Remember account** |
| --- | --- |
| **Description** | This use case allows user to be logged into the system without retyping account information. |
| **Actor** | Normal User, Silver User, Gold User, Administrator |
| **Trigger** | * Click on “Remember me” checkbox |
| **Pre-condition** | * Must be connected to the network * Has enter already username, password * Click on “Remember me” checkbox |
| **Post-condition** | * Account information is saved in database. * In the next login, do not need to validate data * When log out, account information is automatically entered in textboxes |

**Activity Flow**

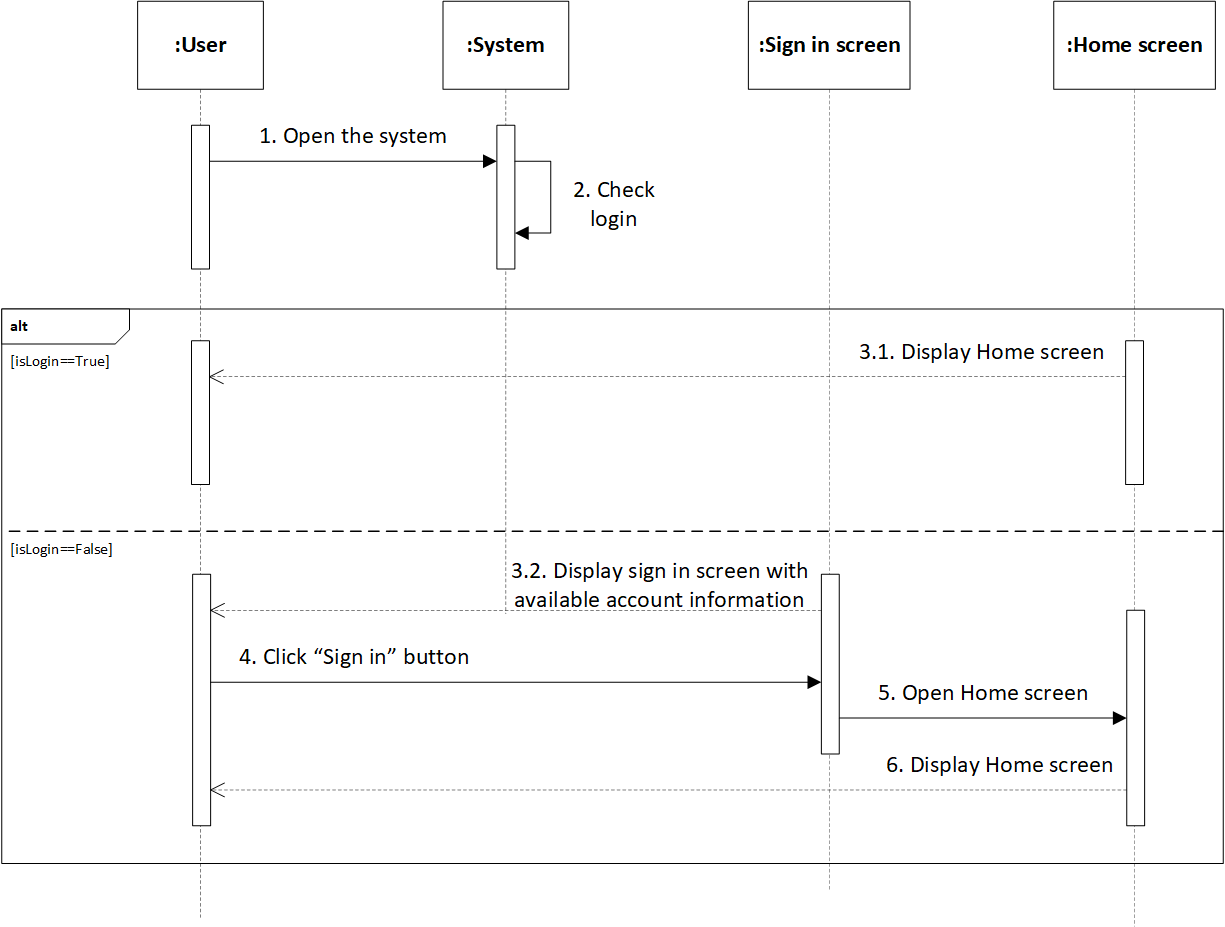


*Figure 8: Activity Flow - Remember account (1)*



*Figure 9: Activity Flow - Remember Account (2)*

**Sequence diagram**



*Figure 10: Sequence Diagram - Remember account*

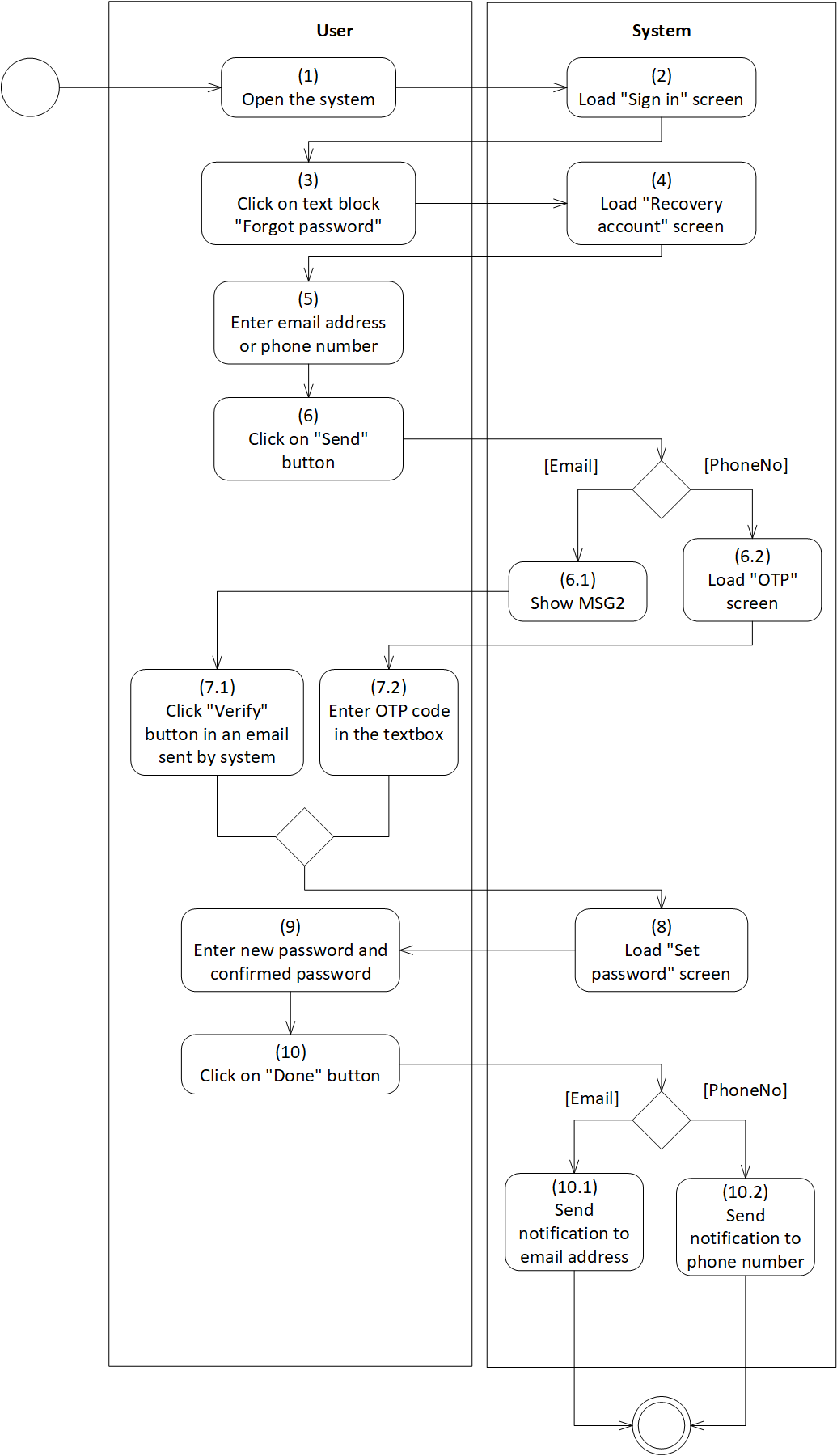
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Check login rules**:  The system will check the login status as the following:  If [isLogin] = True, display the “Home” screen.  If [isLogin] = False, load the “Sign in” screen. |
| *(2)* | *BR2* | **Load “Sign in” Form rules:**  The system will load “Sign in” form with username and password automatically entered.  When User clicks on “Sign in” button, the system will load “Home” screen. |
|  |  | **Sign in Form**:  Information in “Sign in” form as the template below:   | Username | [Username] of current user | | --- | --- | | Password | [Password] of current user | |

# Forgot password

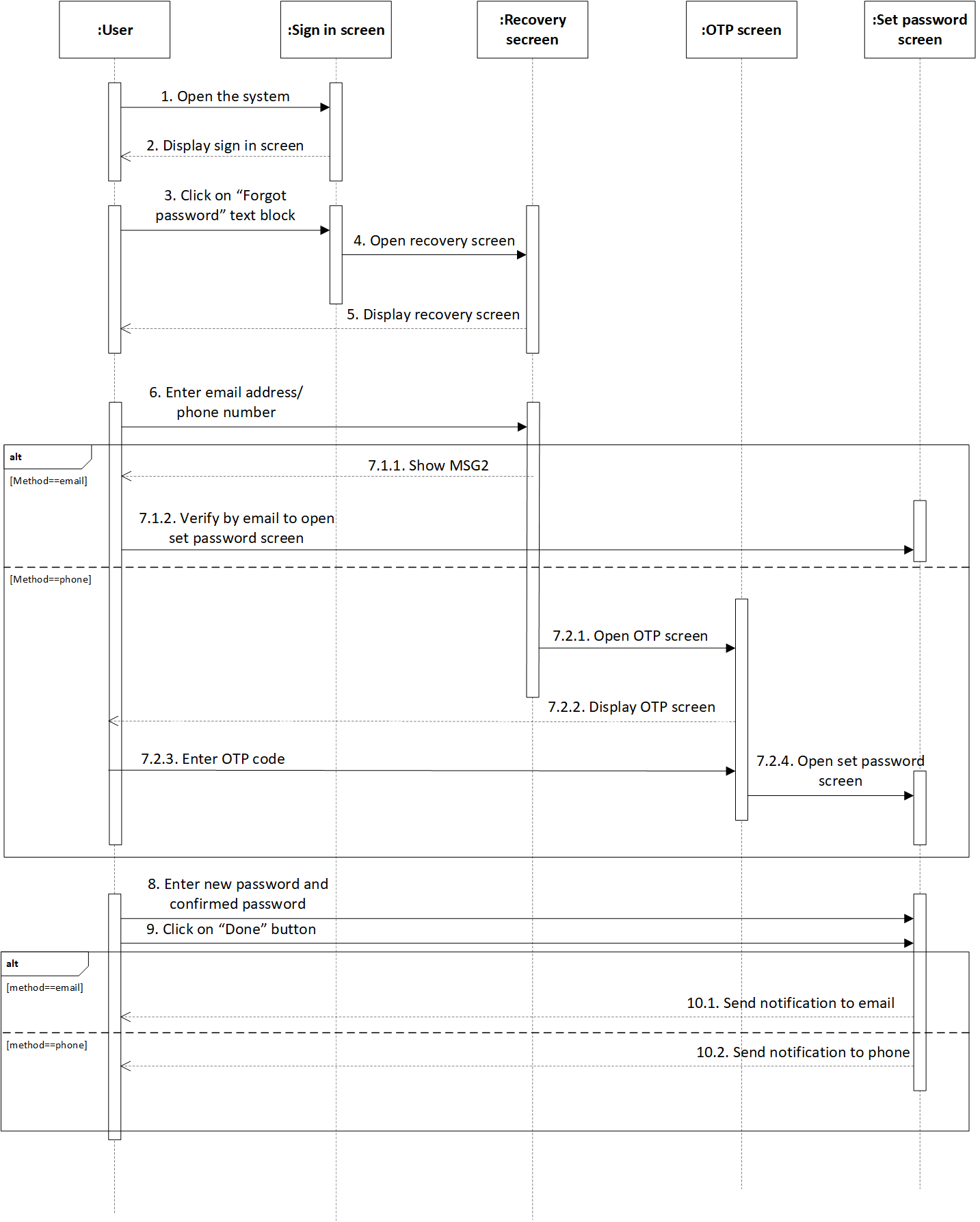
| **Name** | **Forgot password** |
| --- | --- |
| **Description** | This use case allows user to set a new password via verification code. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Forgot password” text block. |
| **Pre-condition** | * Must be connected to the network * Open the application * Click on “Forgot password” text block |
| **Post-condition** | * The password is changed * Verification messages are sent via email |

**Activity Flow**



*Figure 11: Activity Flow - Forgot password*

**Sequence Diagram**



*Figure 12: Sequence Diagram - Forgot password*

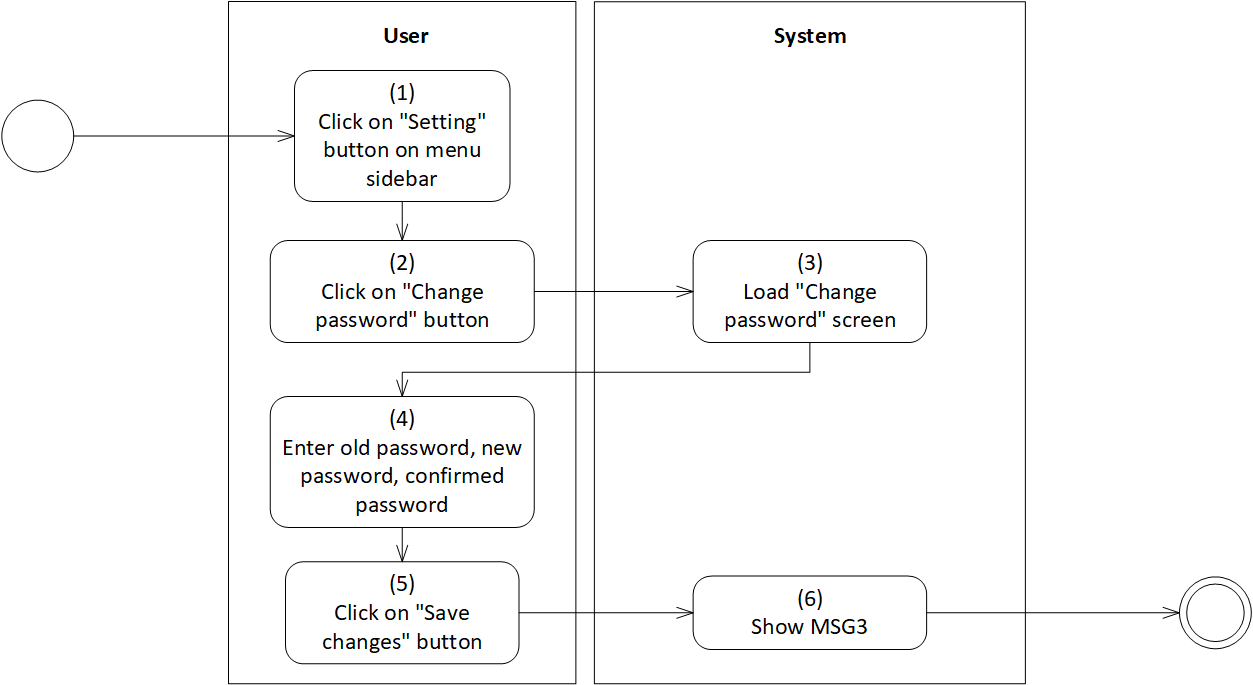
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Sign in” Form rules**:  The system loads “Sign in” form (Refer to “Sign in” list in “UI Description” file) |
| *(4)* | *BR2* | **Load “Recovery account” Form rules**:  When user clicks on “Forgot password” text block, the system will load “Recovery account” form |
| *(5)* | *BR3* | **Input checking rules:**  User enters an email address or phone number in the textbox of “Recovery account” form. If user fills in information, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” * Else if format of [Textbox Text] is incorrect, set [Text Error] = “Please enter in a correct format” |
|  |  | **Recovery account Form**:  Fill information in “Recovery account” form as the template below:   | Email | * Textbox * Check valid email by using a regular expression @"^[^@\s]+@[^@\s]+\.[^@\s]+$" | | --- | --- | | Phone No | * Textbox * Length of [Phone No] >=10 | |
| *(6)* | *BR4* | **Send verification code rules:**  When user clicks on “Send” button, the system will send verification code to user as the following:   * If [Textbox Name] = “Email”, * An email will be sent to [Textbox Text] of email. * The system will load MSG2 * If [Textbox Name] = “Phone No”, * SMS will be sent to [Textbox Text] of Phone No. * The system will load “OTP” screen |
| *(7)* | *BR5* | **Input checking rules:**  User enters verification code in the textbox of verification form. When user clicks on “Confirm” button:   * If [Text] != [OTP Code], * The system will load MSG0 * The system send OPT code again * Else, the system will load “Set password” screen |
| *(9)* | *BR6* | **Input checking rules:**  User enters new password and confirms password in the textboxes of “Set password” form. If user fill in information correctly, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” * Else if [Confirmed Password] != [New Password], set [Text Error] = “New password and Confirmed password must be the same” |
|  |  | **Set password Form**:  Fill information in “Set password” form as the template below:   | New password | * Textbox | | --- | --- | | Confirmed password | * Textbox * [Confirmed Password] = [New Password] | |
| *(10)* | *BR7* | **Send confirmation rules:**  When user clicks on “Done” button of “Set password” form, the system will send password change confirmation as the following:   * If [Textbox Name] = “Email”, * An email will be sent to [Textbox Text] of Email. * If [Textbox Name] = “PhoneNo”, * SMS will be sent to [Textbox Text] of PhoneNo. |

# Change password

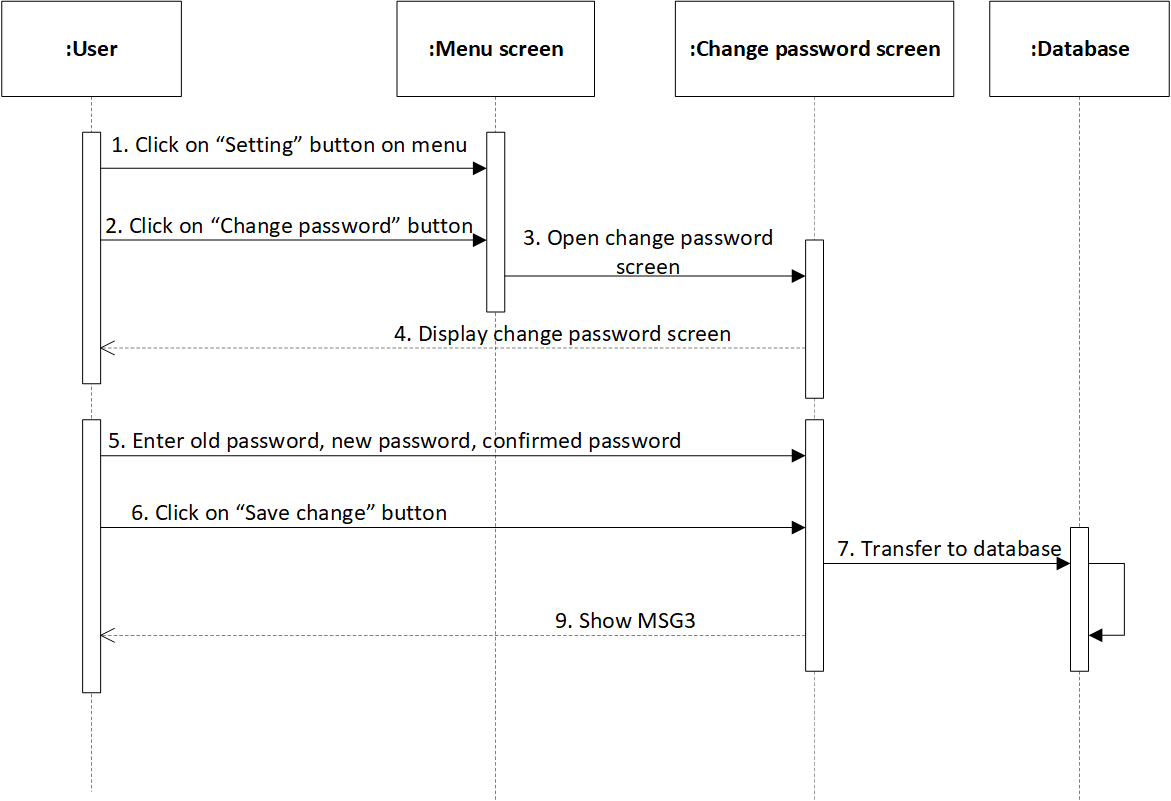
| **Name** | **Change password** |
| --- | --- |
| **Description** | This use case allows user to change password. |
| **Actor** | Normal User, Silver User, Gold User. |
| **Trigger** | * Open the application * Click on “Change password” button. |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Setting” button on menu * Click on “Change password” button |
| **Post-condition** | * Show MSG3 * The password is changed |

**Activity Flow**



*Figure 13: Activity Flow - Change password*

**Sequence Diagram**



*Figure 14: Sequence Diagram - Change password*

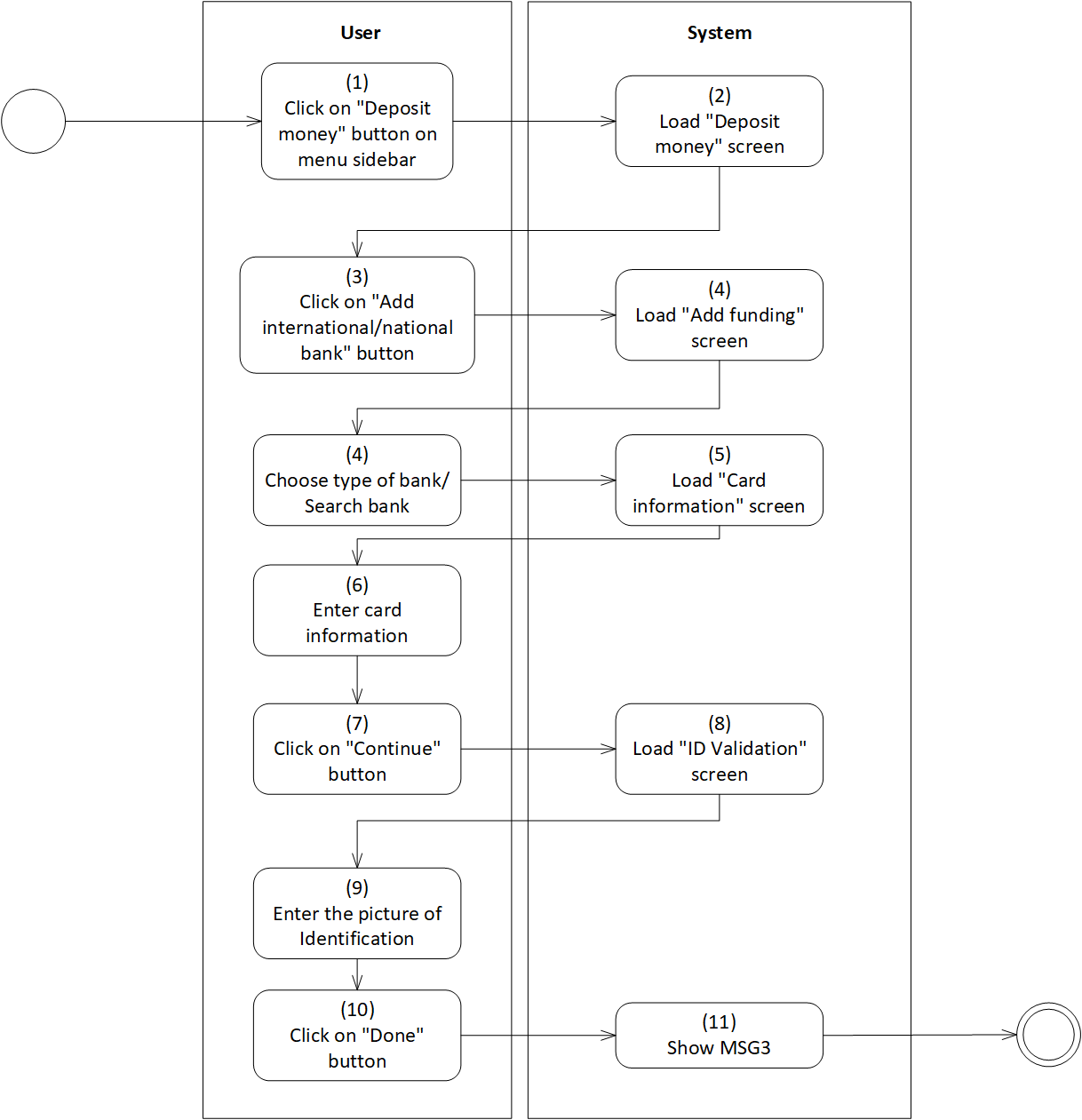
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | Load “Change password” Form rules:  When user clicks on “Setting” button, then clicks on “Change password” button, the system loads “Change password” form. (Refer to “Change password” list in “UI Description” file) |
| *(4)* | *BR2* | **Input checking rules:**  User enters old password, new password and confirmed password in the textboxes of “Change password” form. If user fills in information correctly, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” * Else if [Confirmed Password] != [New Password], set [Text Error] = “New password and Confirmed password must be the same” |
|  |  | **Change password Form**:  Fill information in “Set password” form as the template below:   | Old password | Get [Password] of current usage. | | --- | --- | | New password | * Textbox * Check valid email by using a regular expression @"^[^@\s]+@[^@\s]+\.[^@\s]+$" | | Confirmed password | * Textbox * [Confirmed Password] = [New Password] | |
| *(5)* | *BR3* | **Save changes rules:**  The system retrieves [Password] and checks whether [Old Password] = [Password].   * If yes, * The system will update [Password] = [New Password] * The system will load MSG3 * If no, * The system will load MSG0 * User enters [Old Password] again |

# Link bank account

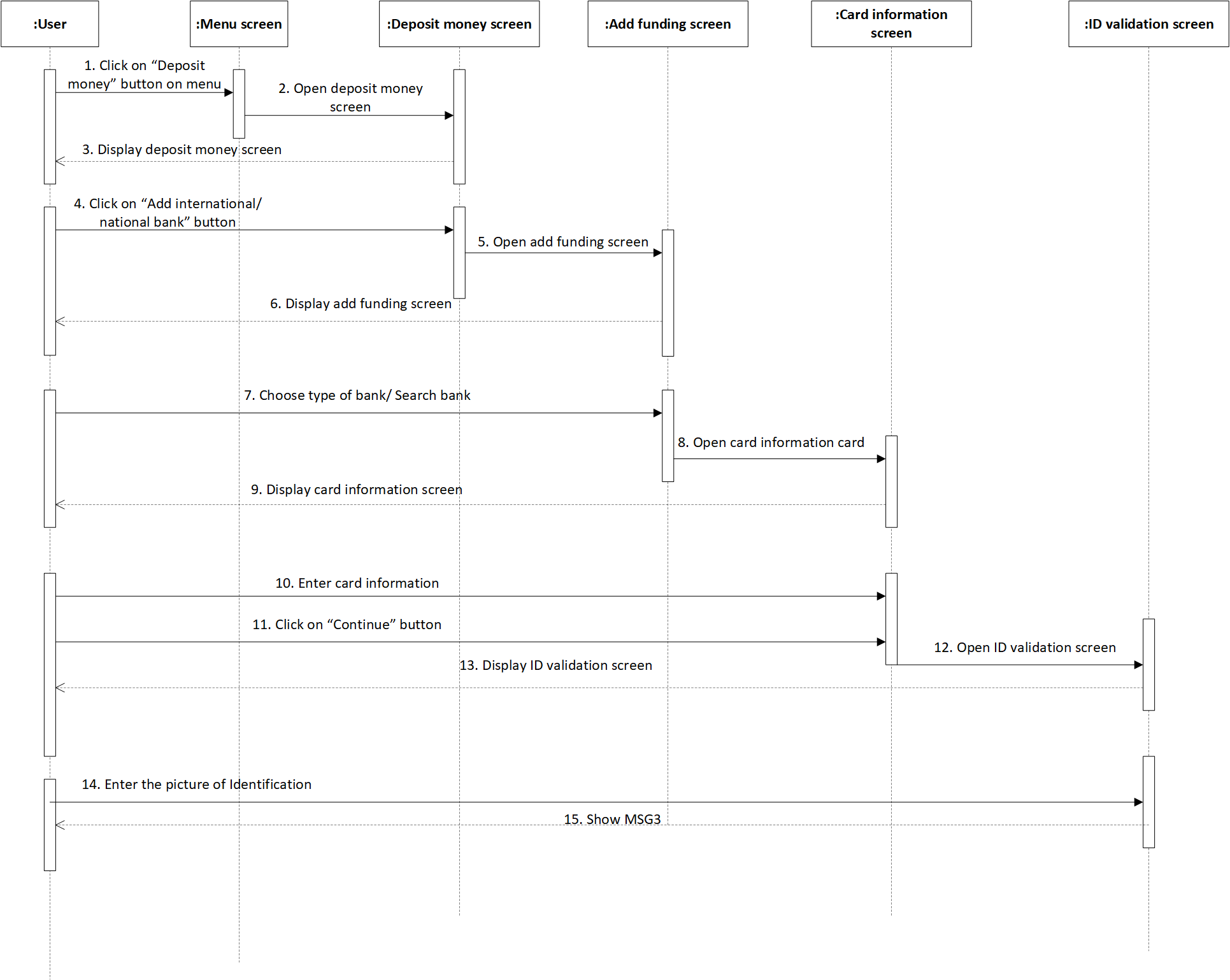
| **Name** | **Link bank account** |
| --- | --- |
| **Description** | This use case allows user to link the application with bank account. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Deposit money” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Deposit money” button on menu |
| **Post-condition** | * Show MSG3 * The bank has just added will be appeared in the source of money. |

**Activity Flow**



*Figure 15: Activity Flow - Link bank account*

**Sequence Diagram**



*Figure 16: Sequence Diagram - Link bank account*

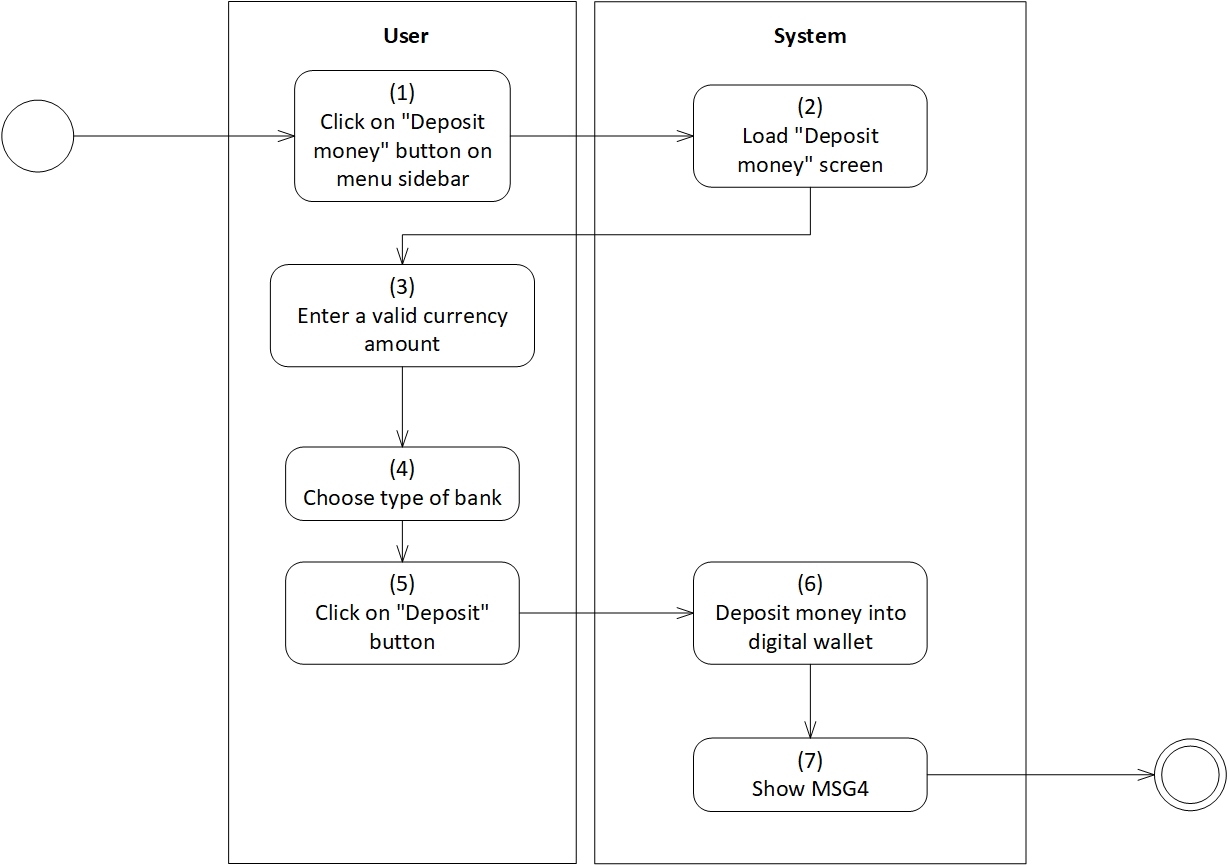
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Deposit money” screen rules**:  When user clicks on “Deposit money” button on the menu sidebar, the system loads “Deposit money” screen. (Refer to “Deposit money” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Add funding” screen rules**:  When user clicks on “Add international/national bank” button of “Deposit money” screen, the system loads “Add funding” screen (Refer to “Add funding” screen in “UI Description” file). |
| *(5)* | *BR3* | **Load “Card Information” Form rules**:  After user chooses the type of bank on the “Add funding” screen, the system loads “Card Information” form (Refer to “Card Information” in “UI Description” file). |
| *(6)* | *BR4* | **Input checking rules:**  User enters card number, card holder and expiry date in the textboxes of “Card Information” form. If user fills in information correctly, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” * Else if format of [Textbox Text] is incorrect, set [Text Error] = “Please enter in a correct format” |
|  |  | **Card Information Form**:  Fill information in “Card Information” form as the template below:   | Card Number | * Textbox * Length of [Card Number] > [Bank.LengthCardNumber] with [Bank] in [Bank List] | | --- | --- | | Card Holder | * Textbox * [Card Holder] is uppercase * [Card Holder] only contains alphabet characters | | Expiry Date | * Textbox * [Expiry Date] = “mm/YYY” | |
| *(8)* | *BR5* | **Load “ID Validation” screen rules**:   * When user click on “Continue” button of “Card Information” form, the system loads “ID validation” screen (Refer to “ID Validation” screen in “UI Description” file) |
| *(9)* | *BR6* | **Input checking rules**:  User uploads the picture of the Identification front and back. When user clicks on “Done” button, the system will check the quality of these two pictures.   * If [Quality Point] > 7, the system loads MSG3 * Else, the system loads MSG15 |

# Deposit money into digital wallet

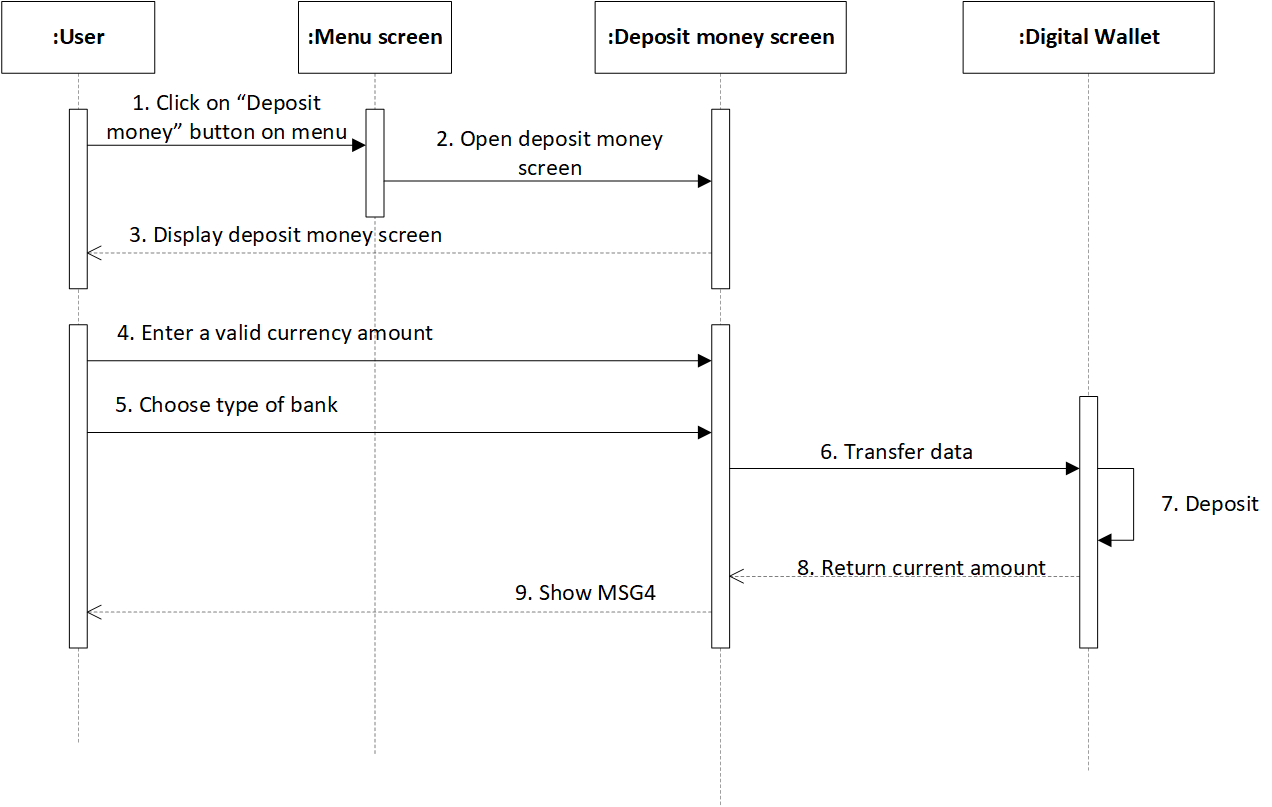
| **Name** | **Deposit money into digital wallet** |
| --- | --- |
| **Description** | This use case allows user to deposit money into its own digital wallet. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Deposit money” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Deposit money” button on menu |
| **Post-condition** | * Show MSG4 * The total money in digital wallet increases equally with the money added |

**Activity Flow**



*Figure 17: Activity Flow - Deposit money into digital wallet*

**Sequence Diagram**



*Figure 18: Sequence Diagram - Deposit money into digital wallet*

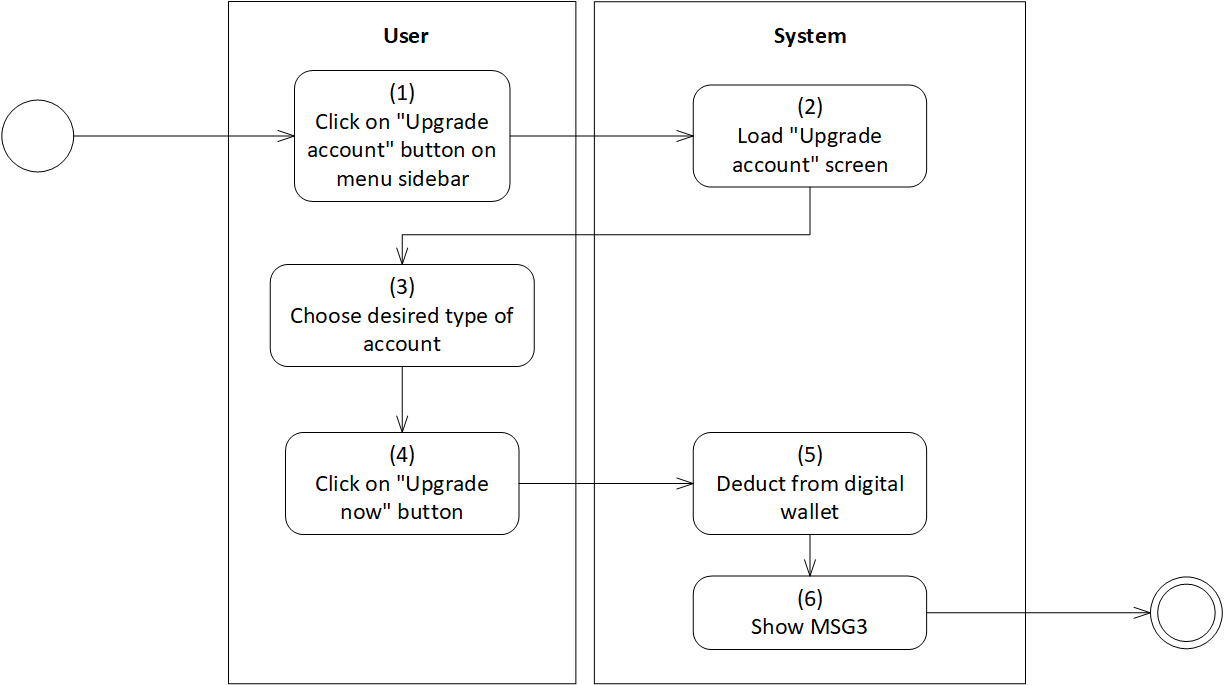
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | Load “Deposit money” screen rules:  When user clicks on “Deposit money” button on menu sidebar, the system loads “Deposit money” screen. (Refer to “Deposit money” screen in “UI Description” file) |
| *(3)* | *BR2* | **Input checking rules:**  User enters the currency amount in the textbox of “Deposit money” form. If user fills in information correctly, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” |
| *(4)* | *BR4* | **Choose bank rules:**  User is only able to choose one of the banks in the list of available banks.   * If no bank has [isChecked] = True, user cannot click on “Deposit” button * If user wants to choose a bank that is different from the banks in list, user need conduct the function “Link bank account” |
| *(6)* | *BR5* | **Deposit money into wallet rules**:  When user click on “Deposit” button of “Deposit money” form, the system will response as the following:   * If [Currency Amount] > [Amount] in bank account, * The system will load MSG0 * User enters [Currency Amount] again. * Else, * The system will update [Wallet Amount] = [Wallet Amount] + [Currency Amount] * The system will load MSG4 |

# Upgrade account

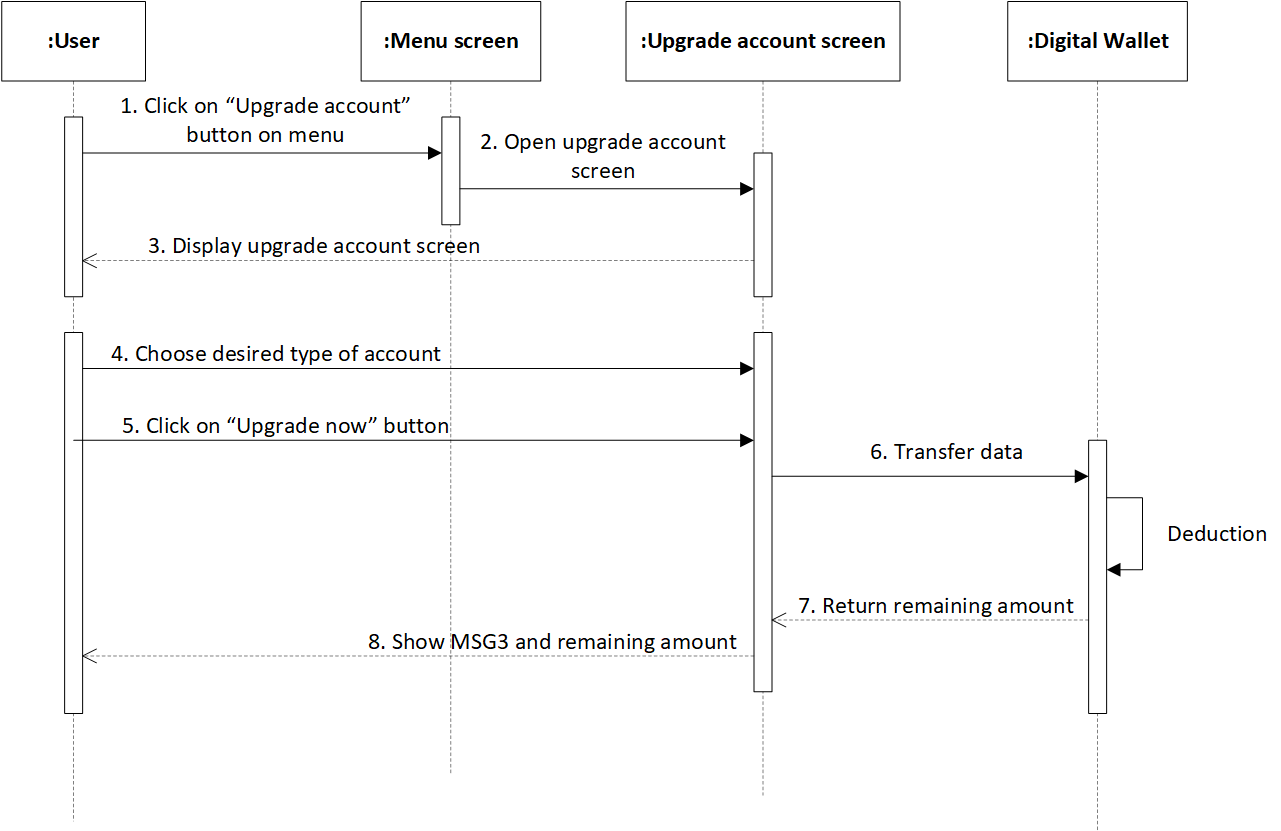
| **Name** | **Upgrade account** |
| --- | --- |
| **Description** | This use case allows user to upgrade the level of user account. |
| **Actor** | Normal User, Silver User. |
| **Trigger** | * Open the application * Click on “Upgrade account” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Update account” button on menu |
| **Post-condition** | * Show MSG3 * The account is upgraded |

**Activity Flow**



*Figure 19: Activity Flow - Upgrade account*

**Sequence Diagram**



*Figure 20: Sequence Diagram - Upgrade account*

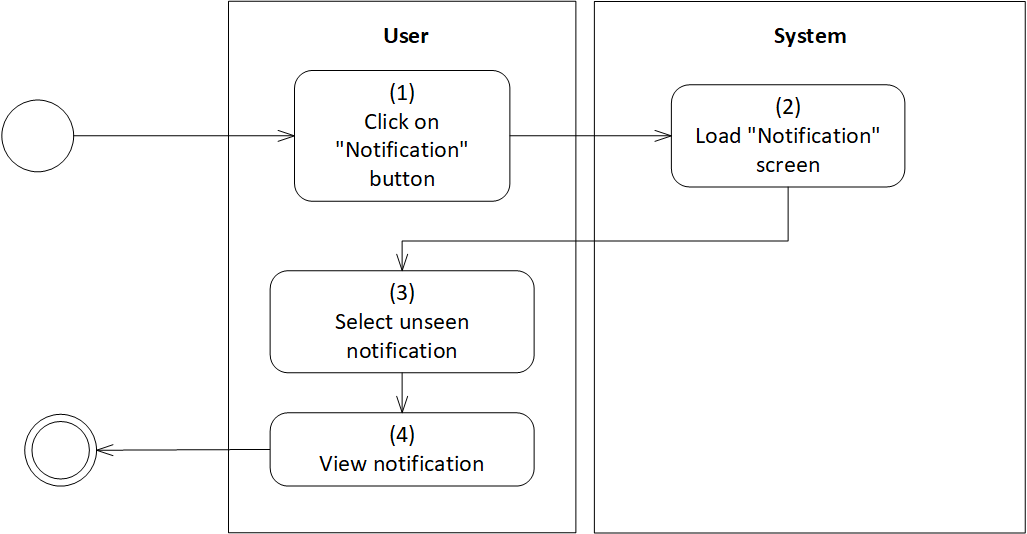
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Upgrade account” screen rules**:  When user clicks on “Upgrade account” button on menu sidebar, the system loads “Upgrade account” screen. (Refer to “Upgrade account” screen in “UI Description” file) |
| *(3)* | *BR2* | **Choose types of account rules:**  Users select a type of account they want to upgrade. Including:  If [Account Type] = “Silver”, update [Current Account Type] = “Silver”   * If [Time] = “1 month”, set [Fee] = 129.000VND/month * Else if [Time] = “6 months”, set [Fee] = 89.000VND/month * Else if [Time] = “12 months”, set [Fee] = 69.000/month   If [Account Type] = “Gold”, update [Current Account Type] = “Gold”   * If [Time] = “1 month”, set [Fee] = 206.000VND/month * Else if [Time] = “6 months”, set [Fee] = 159.000VND/month * Else if [Time] = “12 months”, set [Fee] = 109.000VND/month.   The system will calculate [Total Fee] = [Fee] \* [Time]  The system will calculate [Due Time] = [Current Time] + [Time] |
| *(6)* | *BR3* | Deduct from wallet rules:  When user click on “Upgrade now” button of “Upgrade account” form, the system will response as the following:   * If [Wallet Amount] < [Total Fee], * The system will load MSG0 * The system will load “Deposit money” screen. * Else, * The system will update [Wallet Amount] = [Wallet Amount] - [Total Fee] * The system will load MSG3 |

# Get notification

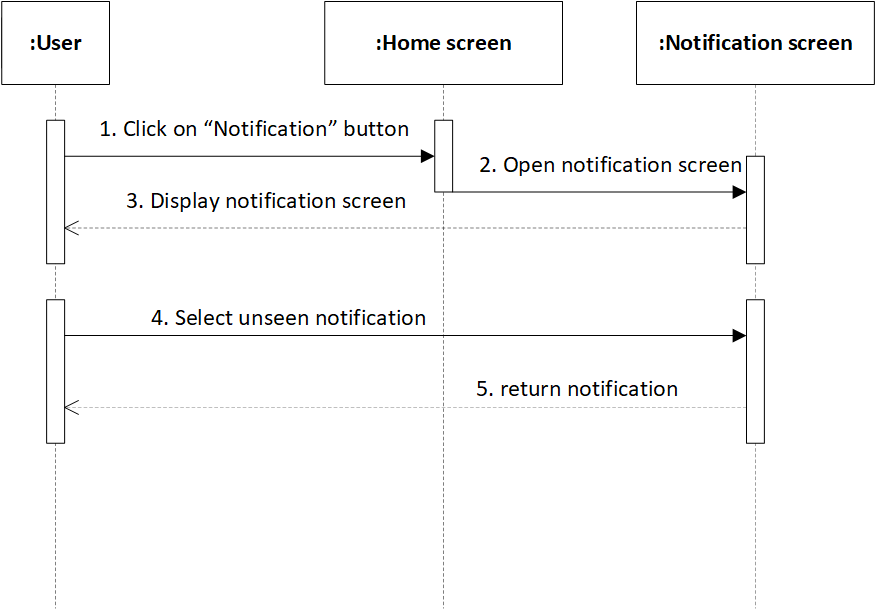
| **Name** | **Get notification** |
| --- | --- |
| **Description** | This use case allows user to get notifications and warnings |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Notification” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the “Home” screen * Click on “Notification” button |
| **Post-condition** | * View the notification |

**Activity Flow**



*Figure 21: Activity Flow - Get notification*

**Sequence Diagram**



*Figure 22: Sequence Diagram - Get notification*

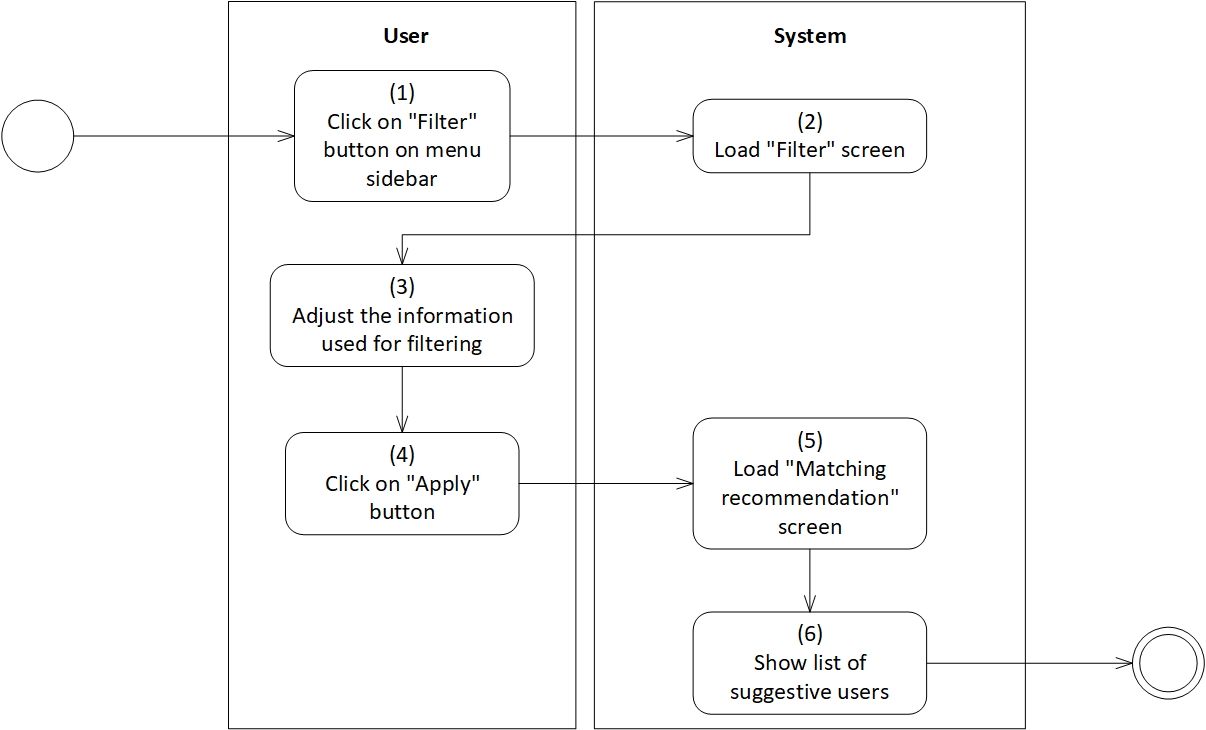
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Notification” rules**:  When user clicks on “Notification” button on “Home” screen, the system loads “Notification”. (Refer to “Notification” list in “UI Description” file) |
| *(3)* | *BR2* | **Notification Form Templates:**  User will receive notifications as the templatebelow:  If [Notification Type] = “Matching Notification”   | Matcher Name | Get [Full Name] of matcher | | --- | --- | | Username | Get [Full Name] of user | | Body | [Body] = “Congratulations” + [Username]  [Body] = [Body] + new line  [Body] = [Body] + [Matcher Name] + “and you have matched successfully!”  [Body] = [Body] + new line  [Body] = [Body] + “Let’s enter the entertainment room to create a chat room right now! Wish both of you have a sweet online conversation” |   If [Notification Type] = “Account Upgrading Notification”   | Current Time | Get [Current Time] of current user | | --- | --- | | Time | Get [Time] user selected while upgrading account | | Current Account Type | Get [Account Type] of current user | | New Account Type | Get [Account Type] user have just upgraded | | Body | [Body] = “ACCOUNT UPGRADATION”  [Body] = [Body] + new 2 lines  [Body] = [Body] + “You have just upgraded your type of account from” + [Current Account Type” + “to” + [New Account Type]  [Body] = [Body] + new line  [Body] = [Body] + “Now, you can use advanced features of our application”  [Body] = [Body] + new line  [Body] = [Body] + “Thank you for choosing our application! Wish you have a good experience.” |   If [Notification Type] = “Warning”   | Username | Get [Username] of current user | | --- | --- | | Body | [Body] = “WARNING”  [Body] = [Body] + new 2 lines  [Body] = [Body] + “Someone has just reported you with administrator group”  [Body] = [Body] + new line  [Body] = [Body] + “We hope” + [Username] + will follow our community rules. If this reporting process continues happening many times, we have to lock your account.” | |

# Filter users

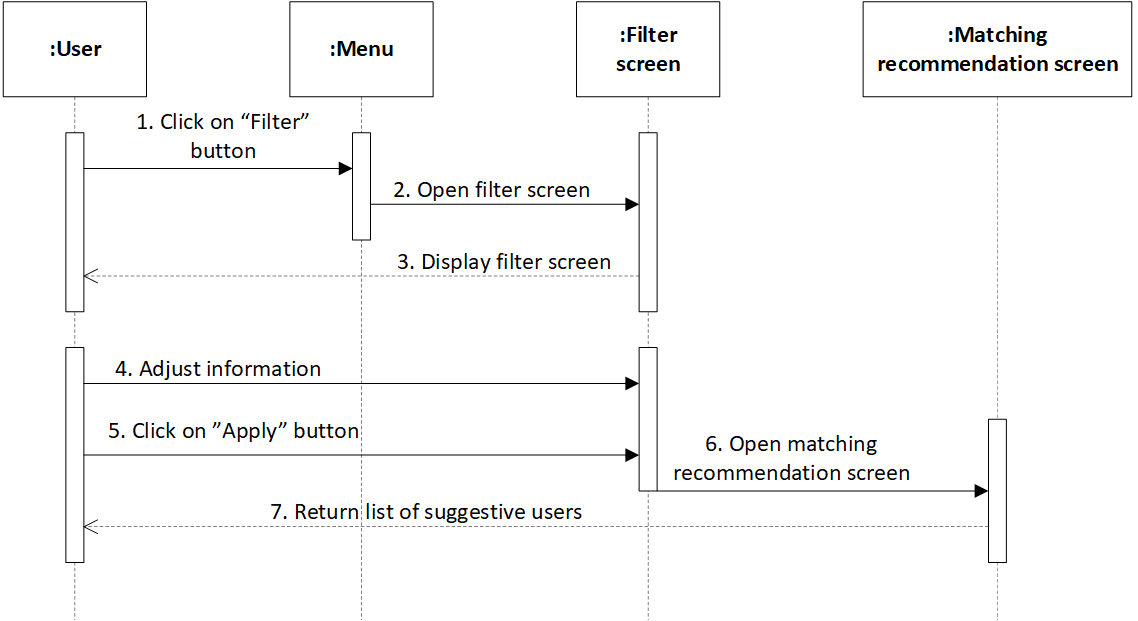
| **Name** | **Filter users** |
| --- | --- |
| **Description** | This use case allows user to filter other users based on information provided to the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Filter” button on the menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Filter” button on the menu |
| **Post-condition** | * Find out the list of compatible users. |

**Activity Flow**



*Figure 23: Activity Flow - Filter users*

**Sequence Diagram**



*Figure 24: Sequence Diagram - Filter users*

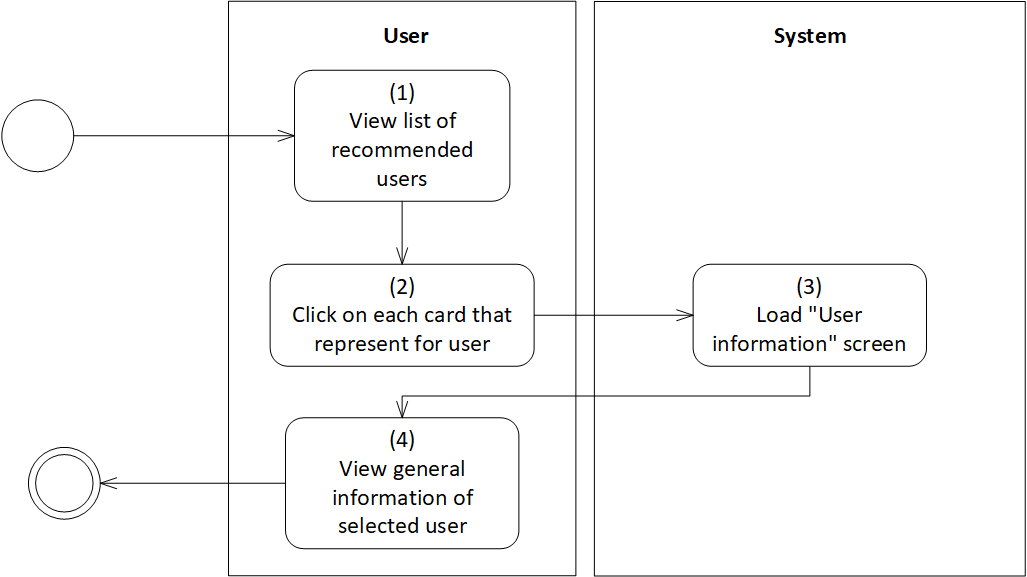
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Filter” Form rules**:  When user clicks on “Filter” button on menu sidebar, the system loads “Filter” form. (Refer to “Filter” list in “UI Description” file) |
| *(3)* | *BR2* | **Adjust information rules:**  User can change information in “Filter” form to find suitable users. If user does not adjust any data field, the system will keep the data remaining. Else, user can adjust information as the following:  If [Account Type] = “Normal”, data fields include:   | Gender | * Radio button * Domain of [Gender] = {“Female”, “Male”, “LGBT”} | | --- | --- | | Age | * Range bar * Domain of [Age] = {18->50} | | Distance | * Range bar * Domain of [Distance] = {0 -> 1260} | | Education | * Radio button * Domain of Education = {“Secondary school”, “High school”, “College”, “University”, “Work”} |  * If [Account Type] = “Silver” | [Account Type] = “Gold”, data fields include:  | Gender | * Radio button * Domain of [Gender] = {“Female”, “Male”, “Other”} | | --- | --- | | Age | * Range bar * Domain of [Age] = {18->70} | | Distance | * Range bar * Domain of [Distance] = {0 -> 1260} | | Education | * Radio button * Domain of Education = “High school”, “College”, “University”, “Work”} | | Profession | * Textbox * User can/ cannot enter [Profession]. | | Hobbies | * [Field] = {“Nature”, “Art”, “Home”, “Sport”, “Other”} * If [Selection] = “Nature”, set [Details] = {“gardening flower/ vegetables”, “feeding and observing the animals”, “fostering animals”} * If [Selection] = “Art”, set [Details] = {“painting”, “sculpture”, “photography”, “pottery”, “hand-made”, “interior decoration”, “singing”, “dancing”, “compose art products”} * If [Selection] = “Home”, set [Details] = {“watching movie”, “cooking”, “playing cards or board games”, “using social media”, “playing video games”, “blogging”, “reading book”} * If [Selection] = “Sport”, set [Details] = {“marathons”, “gym”, “swimming”, “yoga”, “tennis”, “golf”, “football”, “badminton”} * If [Selection] = “Other”, set a textbox for entering data. | |
| *(5)* | *BR3* | **Load “Matching recommendation” screen**:  When user click on “Apply” button of “Filter” form, the system will load “Matching recommendation” screen. This screen shows a list of suggestive users. The number of users recommended by the system will be different at different type of account.   * If [Account Type] = “Normal”, the system displays a list of 5 users per day * Else if [Account Type] = “Silver”, the system displays a list of 15 users per day * Else, the system displays a list of 30 users per day |

# View general user information

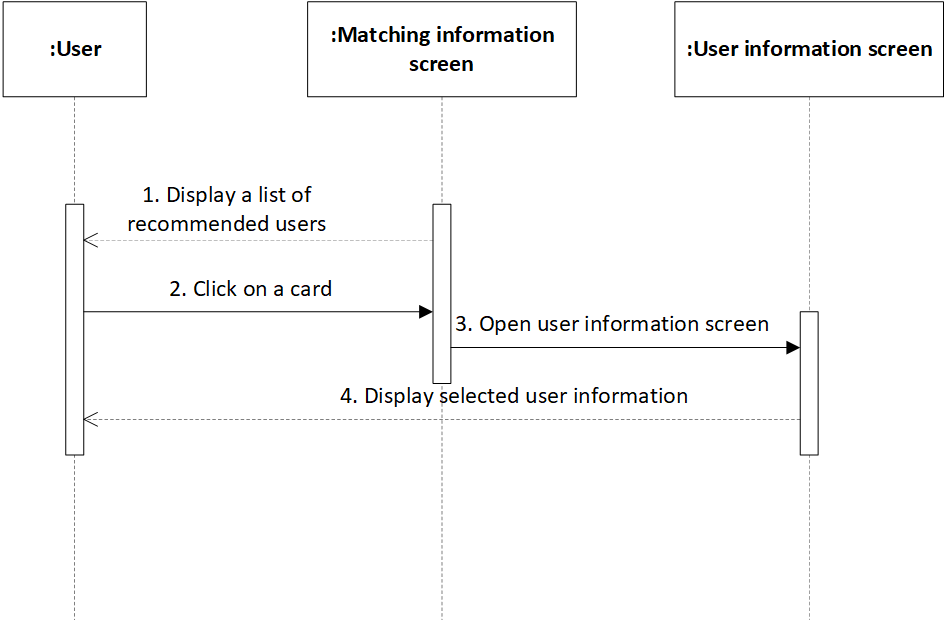
| **Name** | **View general user information** |
| --- | --- |
| **Description** | This use case allows user to view general information of users recommended by the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Filter” button on the menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the matching recommendation screen |
| **Post-condition** | * View general information of each recommended user. |

**Activity Flow**



*Figure 25: Activity Flow - View general user information*

**Sequence Diagram**



*Figure 26: Sequence Diagram - View general user information*

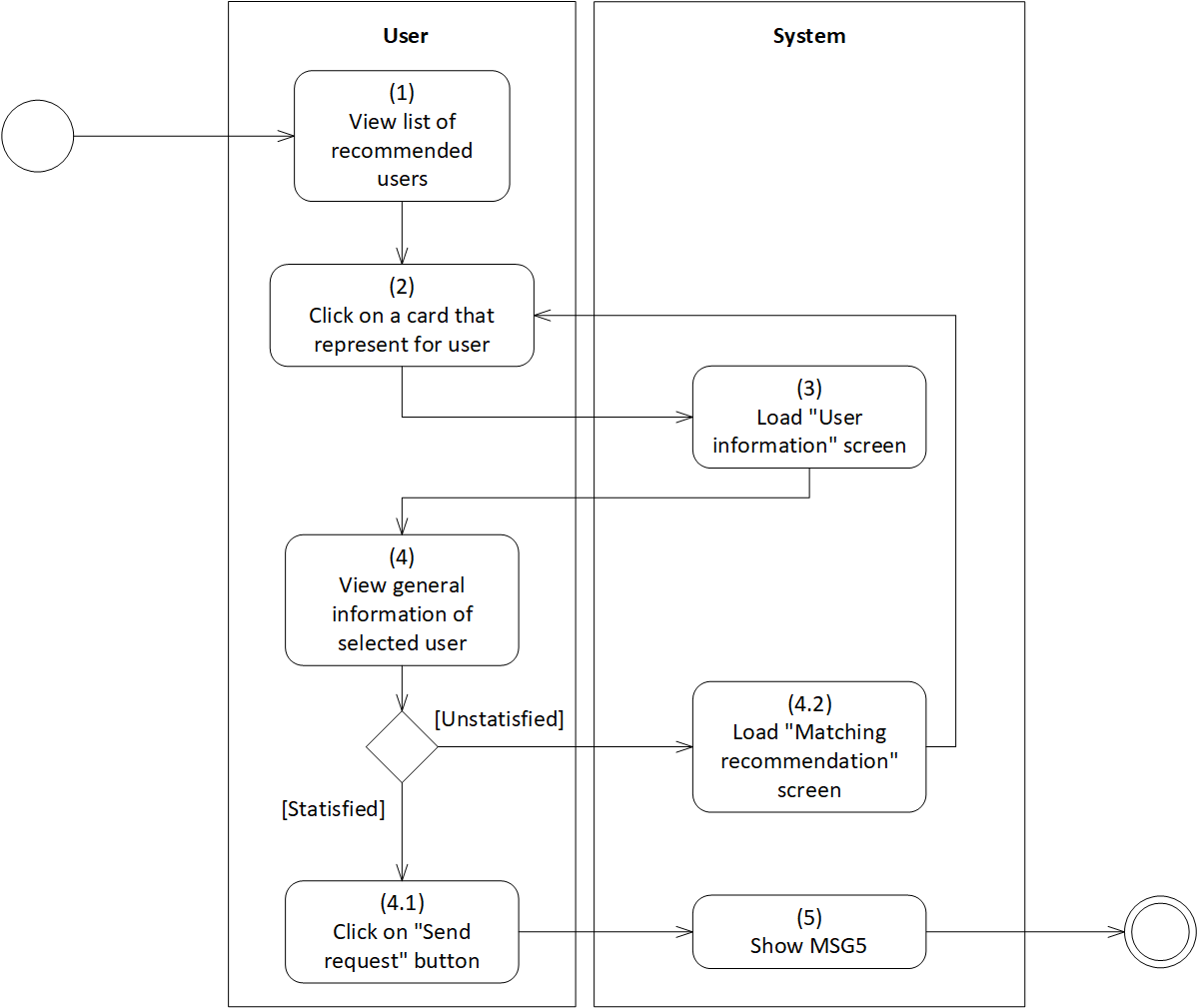
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “User Information” screen rules**:  When user clicks on each card that represents for user, the system loads “User Information” screen. (Refer to “User Information” list in “List Description” file) |
| *(4)* | *BR2* | **User Information Form:**  User can view the general information of selected people as the template below:   | Username | Get [Username] of selected user | | --- | --- | | Gender | Get [Gender] of selected user | | Age | Get [Age] of selected user | | Province | Get [Province] of selected user | | Avatar | Get [Avatar] of selected user | |

# Send matching request

| **Name** | **Send matching request** |
| --- | --- |
| **Description** | This use case allows user to send matching request to other users in the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Filter” button on the menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the matching recommendation screen |
| **Post-condition** | * Succeed in sending matching requests. * Show MSG5 |

**Activity Flow**



*Figure 27: Activity Flow - Send matching request*

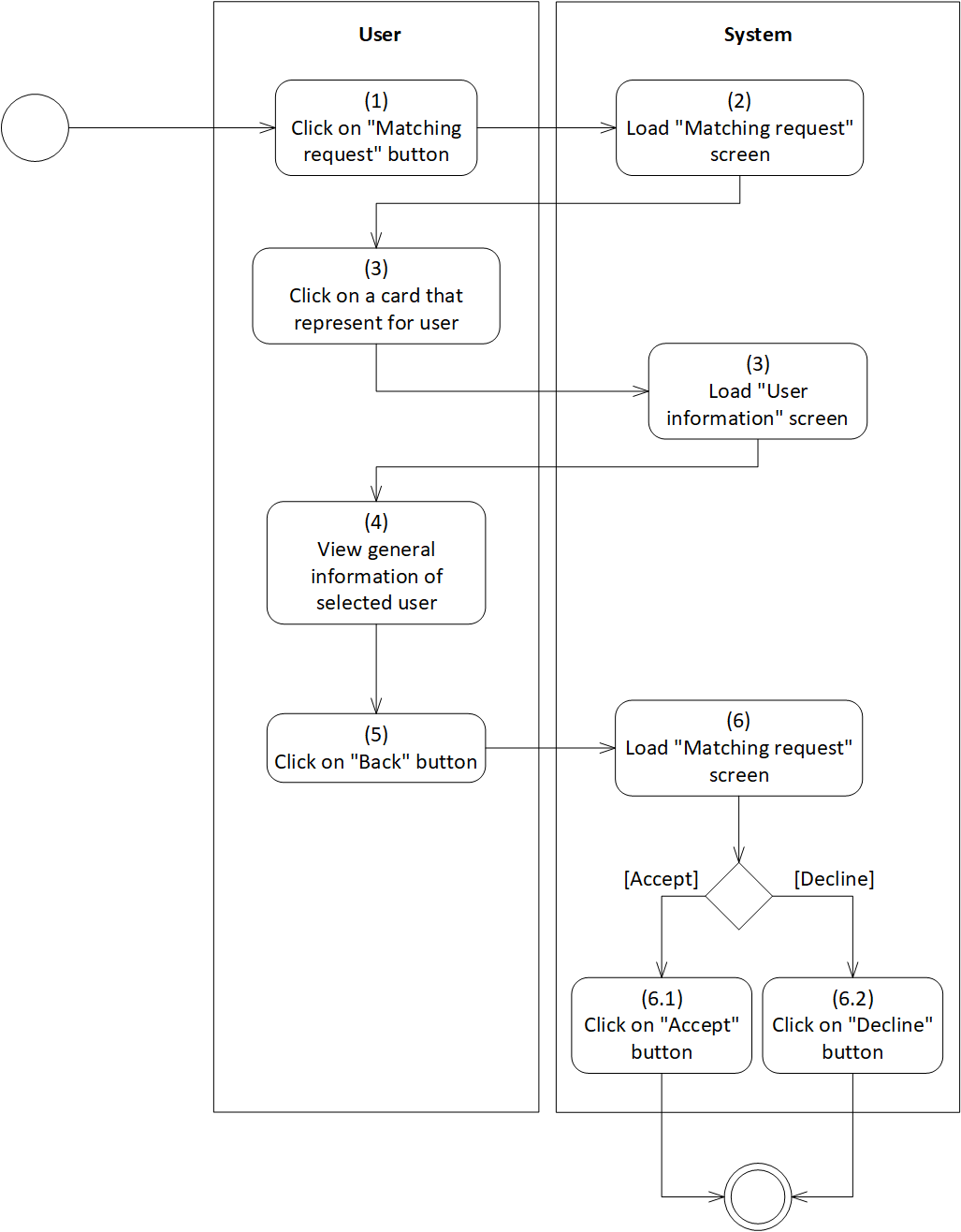
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “User Information” Form rules**:  When user clicks on each card that represents the user, the system loads “User Information” form. (Refer to “User Information” list in “UI Description” file) |
| *(4)* | *BR2* | **User Information Form:**  User can view the general information of selected people as the template below:   | Username | Get [Username] of selected user | | --- | --- | | Gender | Get [Gender] of selected user | | Age | Get [Age] of selected user | | Province | Get [Province] of selected user | | Avatar | Get [Avatar] of selected user | |
|  |  | **Send matching request rules:**   * If user feel satisfied with this selected user, * User can click on “Send request” button * The system will load MSG5 * Else, user can click on “Back” button to load “Matching recommendation” screen again. |

# Receive matching request

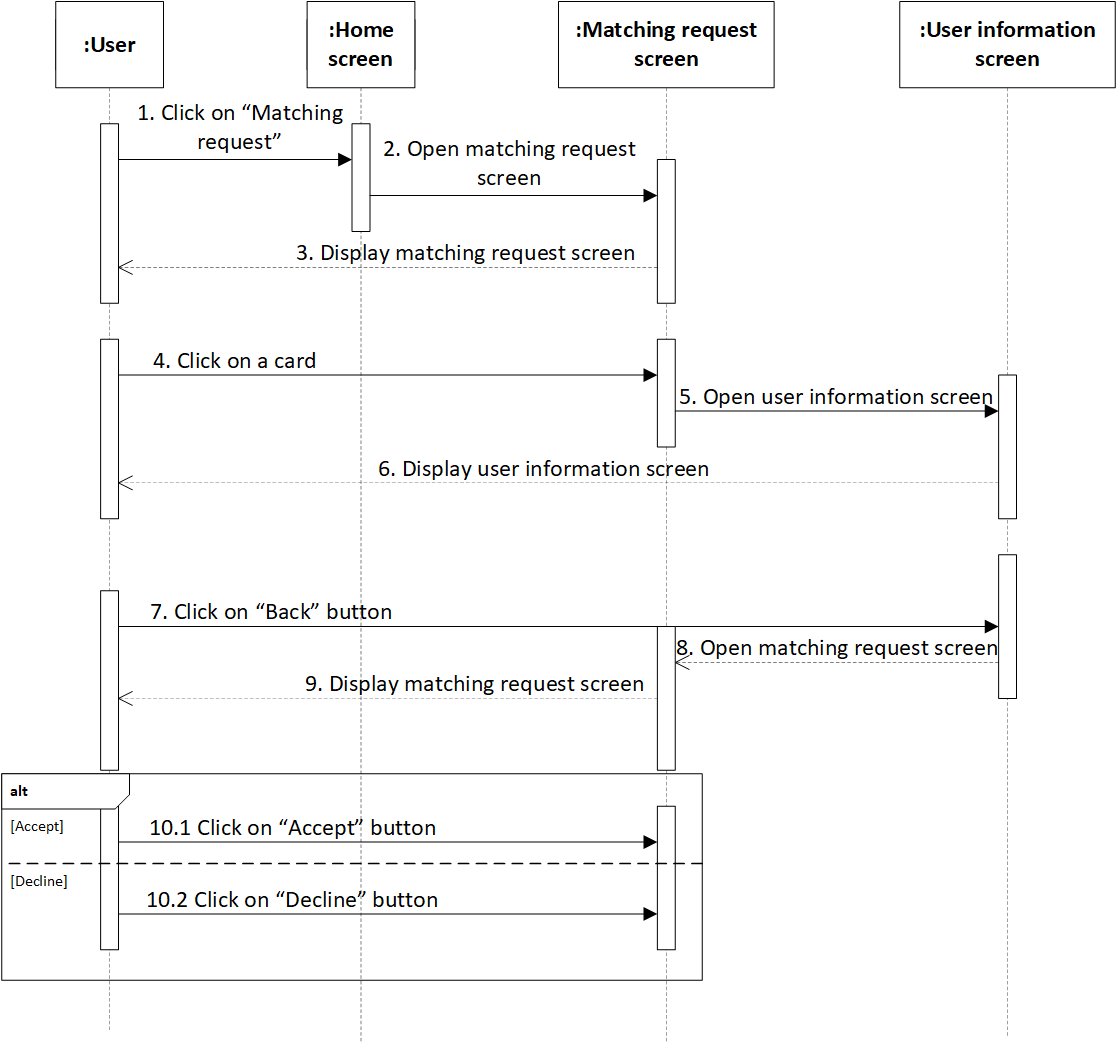
| **Name** | **Receive matching request** |
| --- | --- |
| **Description** | This use case allows user to receive matching request from other users in the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Matching request” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the “Home” screen * Click on “Matching request” button |
| **Post-condition** | * Match successfully * Get matching notification. |

**Activity Flow**



*Figure 28: Activity Flow - Receive matching request*

**Sequence Diagram**



*Figure 29: Sequence Diagram - Send matching request*

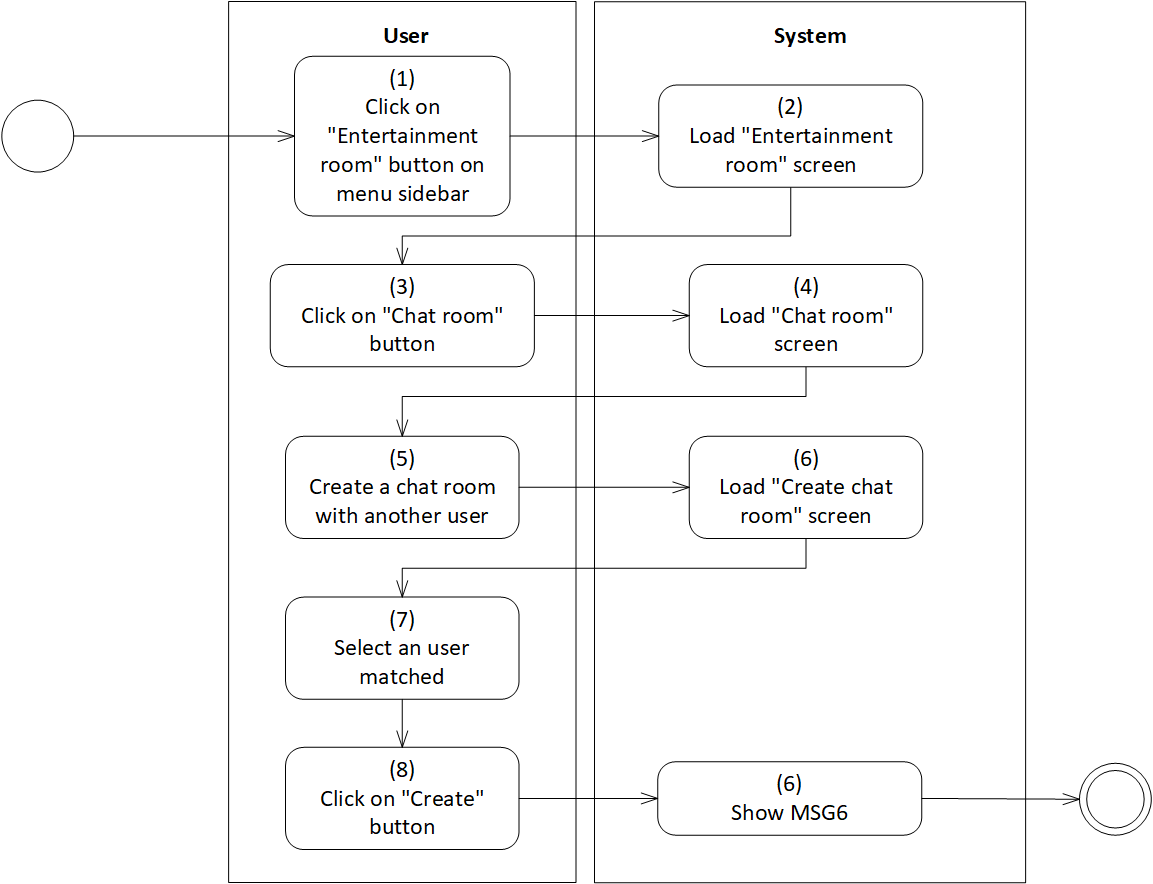
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Matching request” screen rules**:  When user clicks on “Matching request” button, the system loads “Matching request” screen. (Refer to “Matching request” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “User Information” Form rules**:  When user clicks on each card that represents for user, the system loads “User Information” form. (Refer to “User Information” list in “UI Description” file) |
|  |  | **User Information Form:**  User can view the general information of selected people as the template below:   | Username | Get [Username] of selected user | | --- | --- | | Gender | Get [Gender] of selected user | | Age | Get [Age] of selected user | | Province | Get [Province] of selected user | | Avatar | Get [Avatar] of selected user | |
| *(6)* | *BR3* | **Response request rules:**  User clicks on “Back” button to turn back the “Matching request” screen again   * If user feel satisfied with the user sent request, user can click on “Accept” button * Else, user can click on “Decline” |

# Create chat room

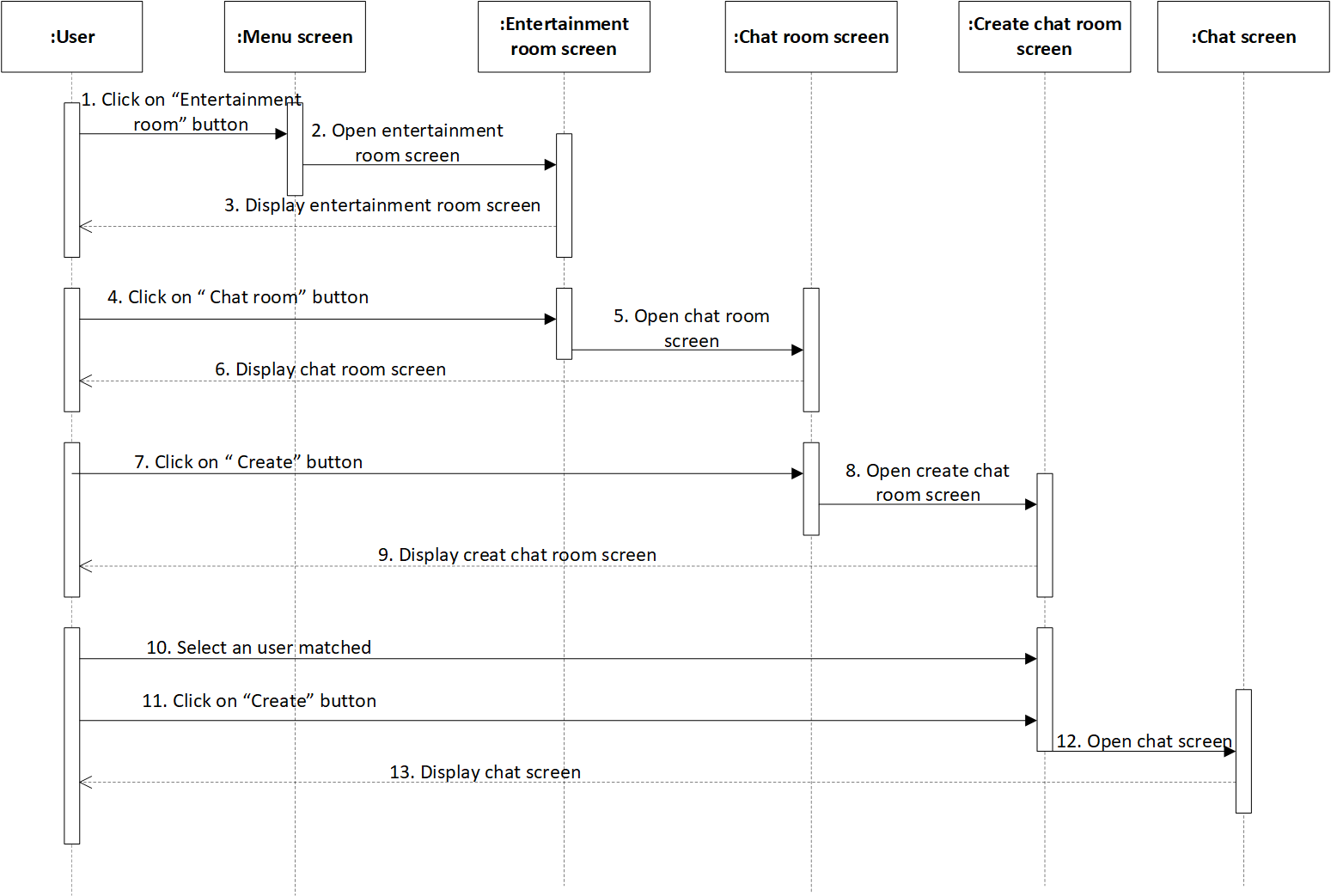
| **Name** | **Create chat room** |
| --- | --- |
| **Description** | This use case allows user to create a chat room with another user in the system. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Entertainment room” button on menu * Click on “Chat room” button * Click on “Create room” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Entertainment room” button * Click on “Chat room” button * Click on “Create room” button |
| **Post-condition** | * Create chat room successfully * Show MSG6. |

**Activity Flow**



*Figure 30: Activity Flow - Create chat room*

**Sequence Diagram**



*Figure 31: Sequence Diagram - Create chat room*

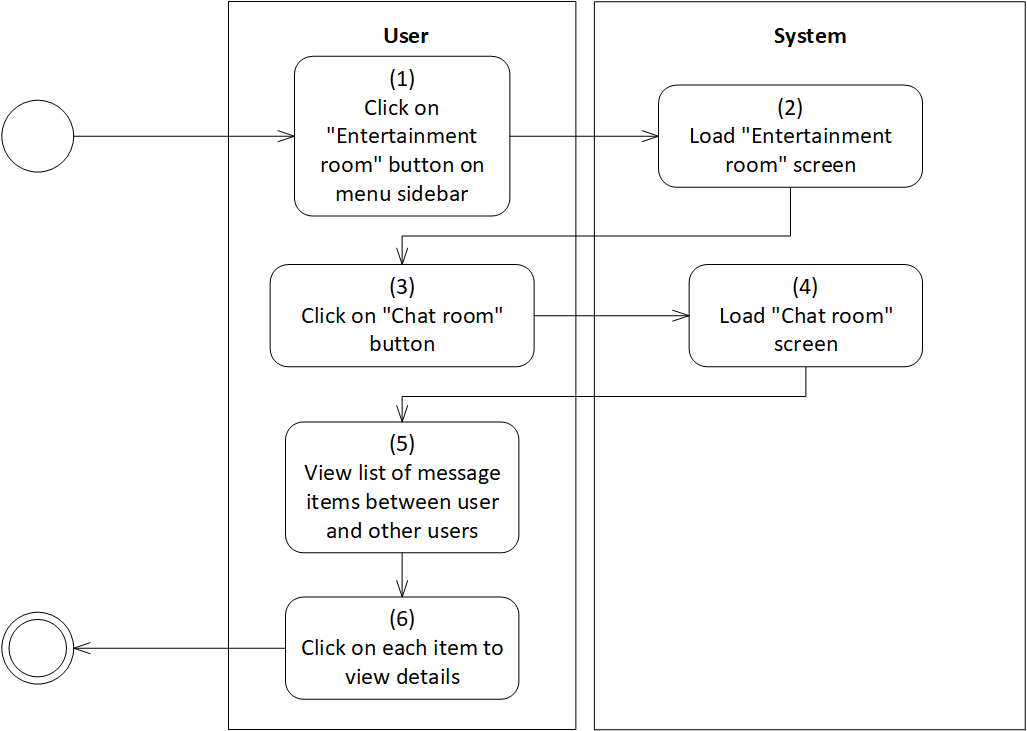
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Entertainment room” screen rules**:  When user clicks on “Entertainment room” button, the system loads “Entertainment room” screen. (Refer to “Entertainment room” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Chat room” screen rules**:  When user clicks on “Chat room” button, the system loads “Chat room” screen. (Refer to “Chat room” screen in “UI Description” file) |
| *(6)* | *BR3* | **Load “Create chat room” screen rules**:  When user clicks on “Create chat room” button, the system loads “Create chat room” screen. (Refer to “Create chat room” screen in “UI Description” file) |
| *(6)* | *BR3* | **Create chat room rules:**  User only select one user in the matching list.   * If [Matching List] = “”, user cannot create chat room * After user selects one user in the matching list, clicks on “Create” button. Then, the system will load MSG6 |

# View chatting history

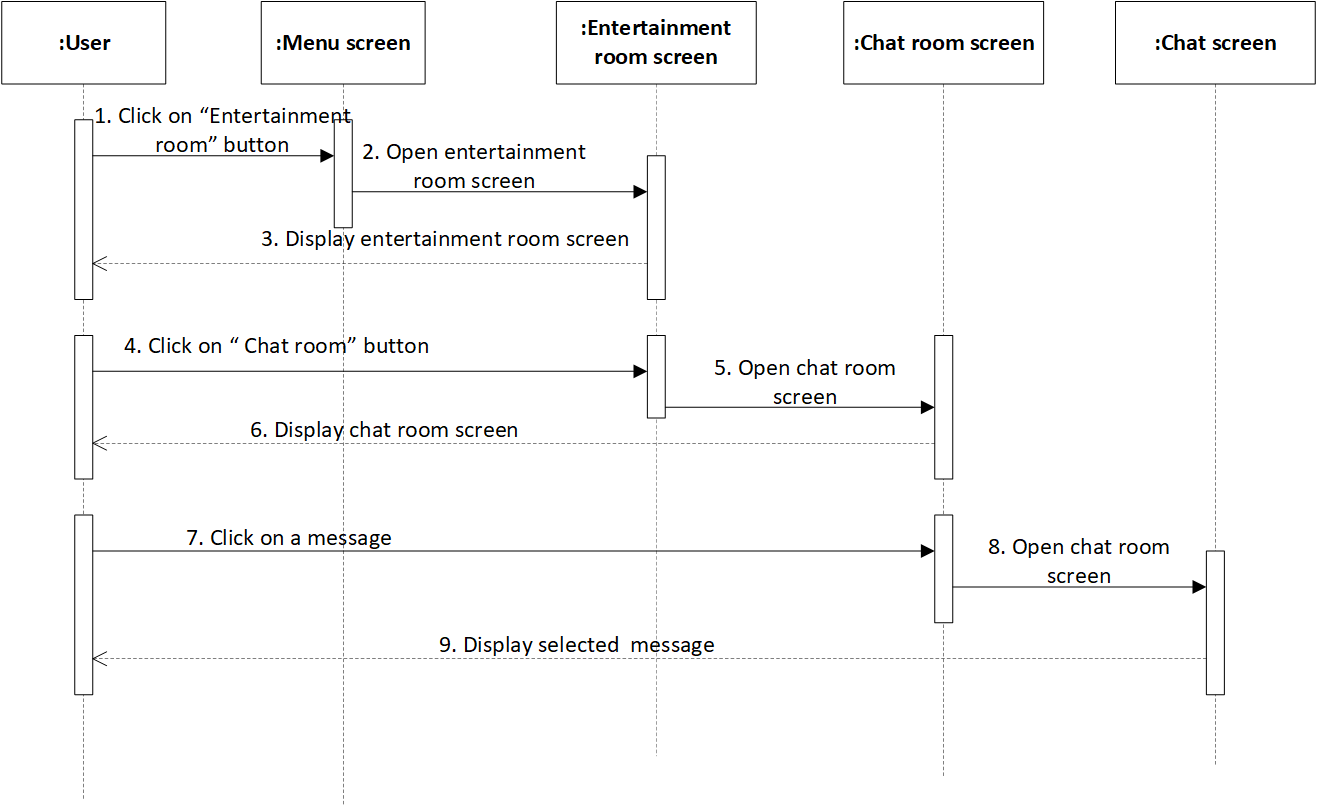
| **Name** | **View chatting history** |
| --- | --- |
| **Description** | This use case allows user to review messages between this user with other users. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Entertainment room” button on menu * Click on “Chat room” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Entertainment room” button * Click on “Chat room” button |
| **Post-condition** | * Review messages |

**Activity Flow**



*Figure 32: Activity Flow - View chatting history*

**Sequence diagram**



*Figure 33: Sequence Diagram - View chatting history*

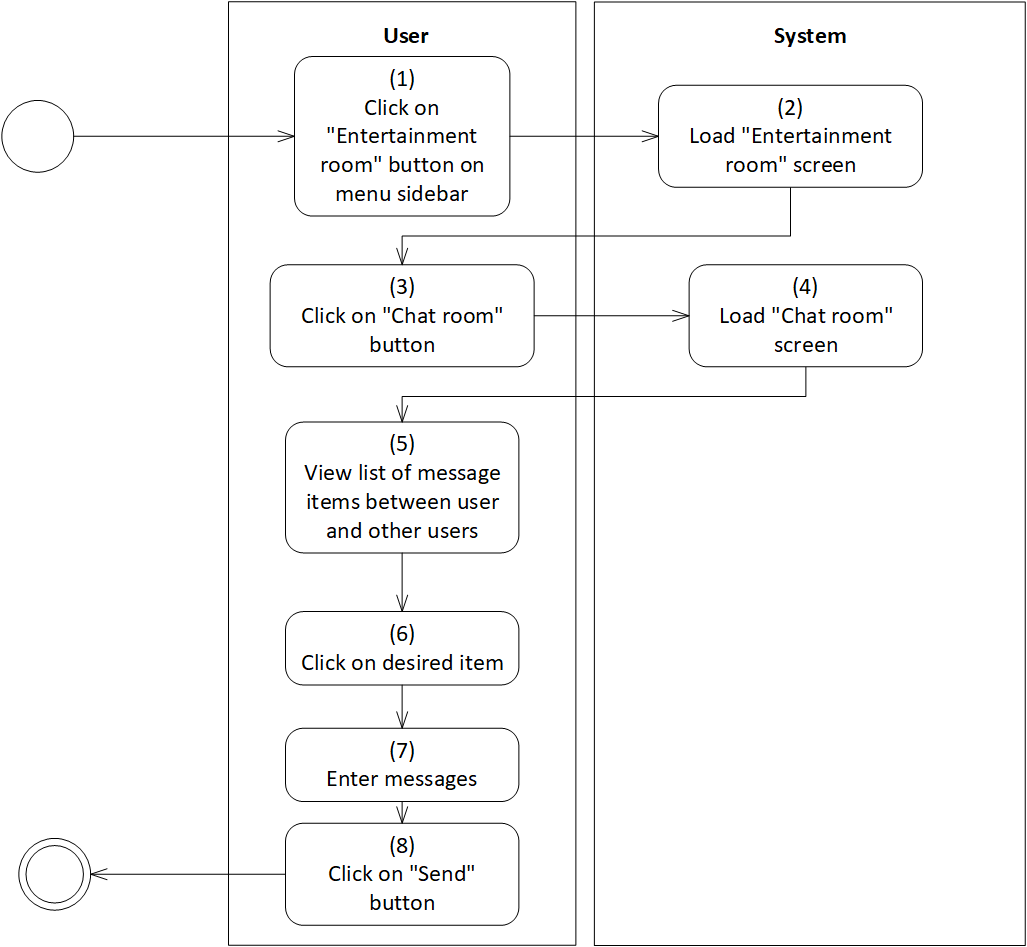
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Entertainment room” screen rules**:  When user clicks on “Entertainment room” button, the system loads “Entertainment room” screen. (Refer to “Entertainment room” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Chat room” screen rules**:  When user clicks on “Chat room” button, the system loads “Chat room” screen. (Refer to “Chat room” screen in “UI Description” file) |
| *(5)* | *BR3* | **List of message items rules:**  In the list message items, each message will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | | Sending Time | Get [Send Time] of the nearest message |  * If [Account Type] = “Normal”, after 5 days, set [Avatar] = “Hidden”, [Username] = “Hidden” * Else, set [Avatar] = “Visibility” and [Username] = “Visibility” |
| *(6)* | *BR4* | **View details rules:**  User can click on each message item to view details. The message details will include:   | Avatar | Get [Avatar] of sender | | --- | --- | | Avatar | Get [Avatar] of receiver | | Username | Get [Username] of sender | | Username | Get [Username] of receiver | | Body | [Body] = [Body] + [Message] of sender + [Message] of receiver | | Message | * Rich Text * User enters message and clicks on “Send” button * Attach [Sending Time] and [Seen Time] | | Sending Time | Get [Send Time] of the nearest message | | Seen Time | Get [Seen Time] of the nearest message |  * If [Account Type] = “Normal”, after 5 days, set [Avatar] = “Hidden”, [Username] of selected user = “Hidden” * Else, set [Avatar] = “Visibility” and [Username] of selected user = “Visibility” |

# Chat

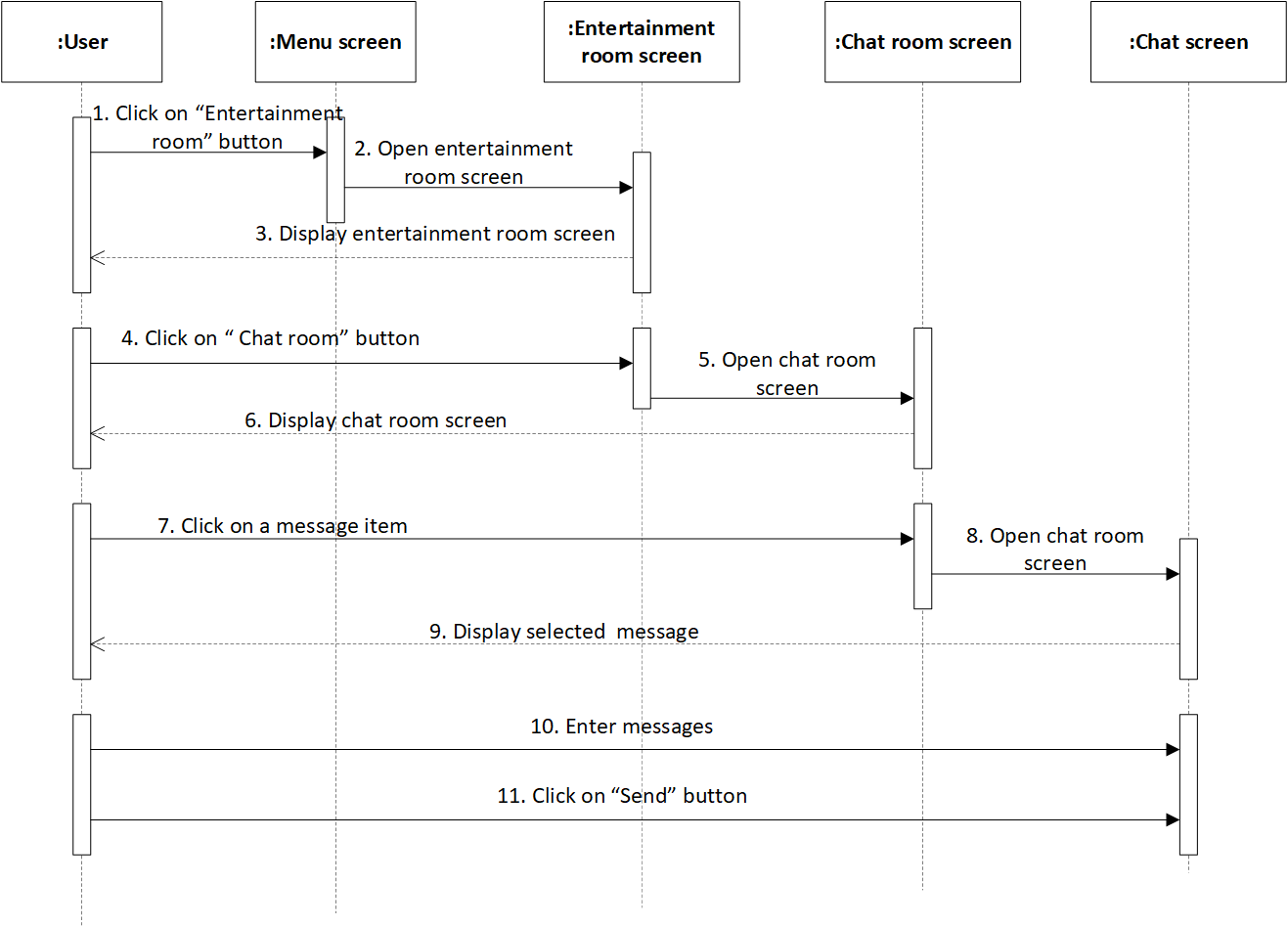
| **Name** | **Chat** |
| --- | --- |
| **Description** | This use case allows user to send messages to other users matched. |
| **Actor** | Normal User, Silver User, Gold User |
| **Trigger** | * Click on “Entertainment room” button on menu * Click on “Chat room” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Entertainment room” button * Click on “Chat room” button |
| **Post-condition** | * Send messages successfully |

**Activity Flow**



*Figure 34: Activity Flow - Chat*

**Sequence Diagram**



*Figure 35: Sequence Diagram – Chat*

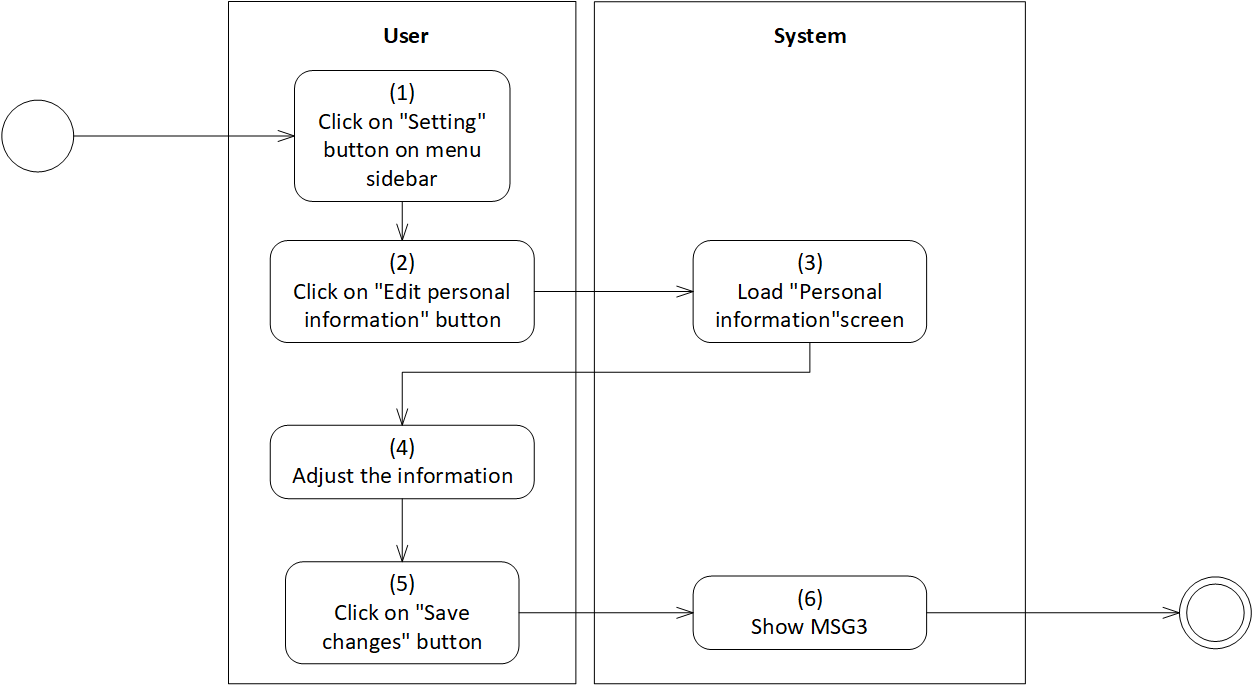
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Entertainment room” screen rules**:   * When user clicks on “Entertainment room” button, the system loads “Entertainment room” screen. (Refer to “Entertainment room” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Chat room” screen rules**:   * When user clicks on “Chat room” button, the system loads “Chat room” screen. (Refer to “Chat room” screen in “UI Description” file) |
| *(5)* | *BR3* | **List of message items rules:**  In the list message items, each message will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | | Sending Time | Get [Send Time] of the nearest message |  * If [Account Type] = “Normal”, after 5 days, set [Avatar] = “Hidden”, [Username] = “Hidden” * Else, set [Avatar] = “Visibility” and [Username] = “Visibility” |
| *(6)* | *BR4* | **View details rules:**  User can click on each message item to view details. The message details will include:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | | Username | Get [Username] of current user | | Body | [Body] = [Body] + [Message] of user 1 + [Message] of user 2 | | Message | * Text box * User enters message and clicks on “Send” button * Attach [Sending Time] and [Seen Time] | | Sending Time | Get [Send Time] of the nearest message | | Seen Time | Get [Seen Time] of the nearest message |  * If [Account Type] = “Normal”, after 5 days, set [Avatar] = “Hidden”, [Username] of selected user = “Hidden” * Else, set [Avatar] = “Visibility” and [Username] of selected user = “Visibility” |
| *(7)* | *BR5* | **Chat rules:**  All types of messages, including:   * If user click on “Emoji” button and [Selection] = [Selected Emoji], set [Textbox Content] = [Emoji] in [Emoji List] with [Emoji] = [Selected Emoji] * If user click on “Picture” button and select a picture from user device, set [Textbox Content] = [Selected Image] * If user enter [Text], set [Textbox Content] = [Text]   In addition, [Textbox Content] = [Text] + [Selected Emoji]  Each type of account will have different rules for chatting duration.   * If [Account Type] = “Normal”, set [Duration] = 10 minutes/user * If [Account Type] = “Silver”, set [Duration] = 30 minutes/user * If [Account Type] = “Silver”, set [Duration] is unlimited. |

# Edit personal information

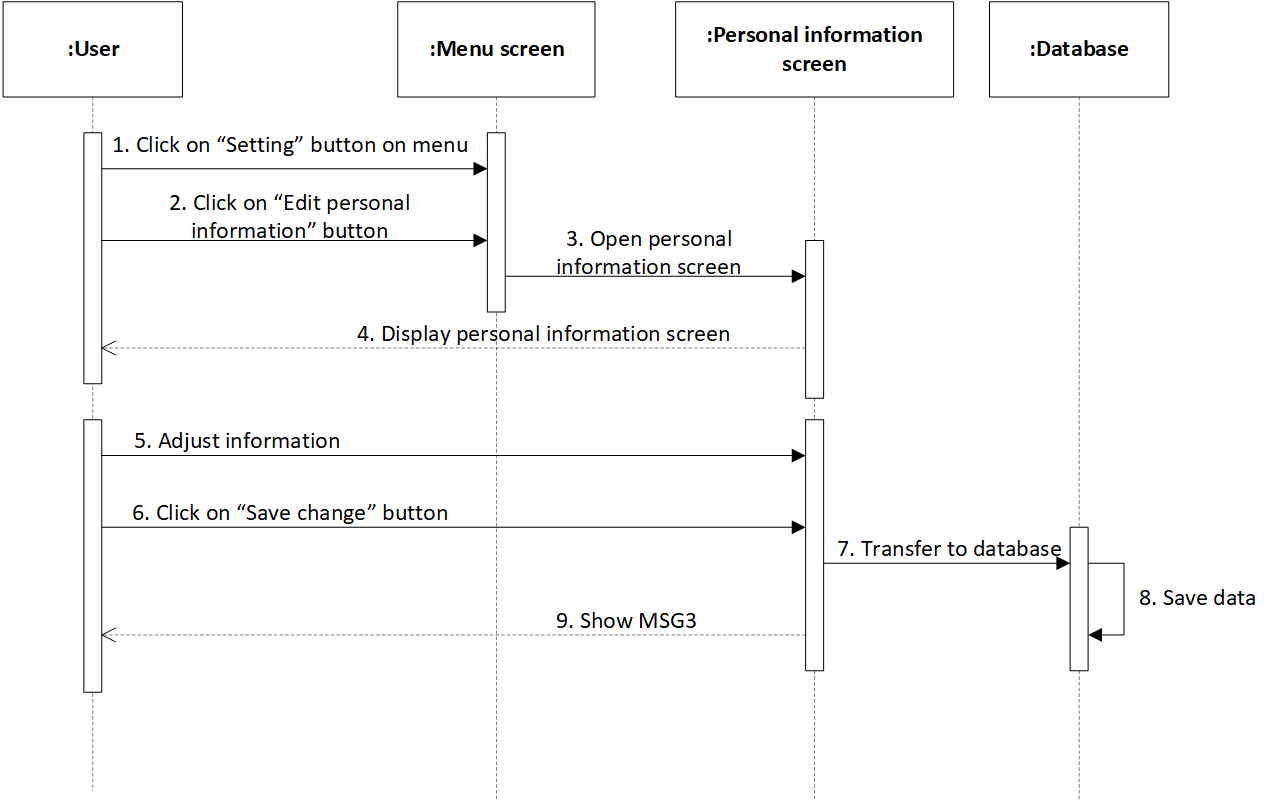
| **Name** | **Edit personal information** |
| --- | --- |
| **Description** | This use case allows user to edit personal information. |
| **Actor** | Normal User, Silver User, Gold User. |
| **Trigger** | * Open the application * Click on “Edit personal information” button. |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Setting” button on menu * Click on “Edit personal information” button |
| **Post-condition** | * Show MSG3 * The personal information is changed |

**Activity Flow**



*Figure 36: Activity Flow - Edit personal information*

**Sequence Diagram**



*Figure 37: Sequence Diagram - Edit personal information*

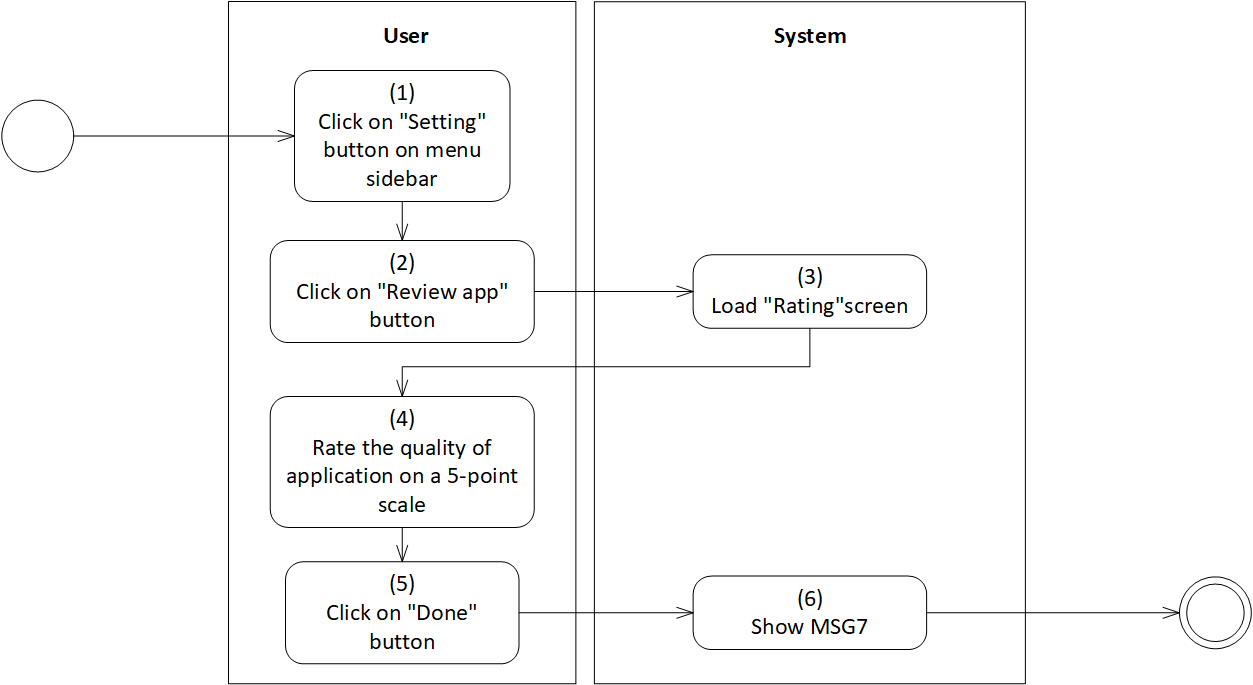
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “Personal Information” Form rules**:   * When user clicks on “Setting” button on menu sidebar, then clicks on “Edit personal information” button, the system loads “Personal Information” form. (Refer to “Personal Information” list in “UI Description” file) |
| *(4)* | *BR2* | **Adjust information rules:**  User adjusts personal information in the textboxes of “Personal Information” form. If user enters according to the accurate format, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” * Else If type of [Textbox Text] != [Specified Data Type], set [Text Error] = “Please enter”+“<<Field Name>> + “as”+ <<Specified Data Type>> * Else If [Textbox Text] do not contains all types of character, set [Text Error] = “<<Field Name>>” + “must have” + “<<List of character types>>” * Else if length of [Textbox Text] is smaller than specified length, set [Text Error] = “Please enter” + “<<Specified Length>>” + “character” |
|  |  | **Personal Information Form**:  Fill information in “Personal Information” form as the template below:   | Full name | * Textbox * Length [Full name] >= 8 | | --- | --- | | Date of birth | * Textbox * [DOB]= “dd-mm-YYYY” | | Gender | * Radio button * Domain of [Gender] = {“Female”, “Male”, “LGBT”} | | Phone No | * Textbox * Length [Phone No] >= 10 | | Email | * Textbox * Check valid email by using a regular expression: @"^[^@\s]+@[^@\s]+\.[^@\s]+$" | | Address | * Combo box includes name of provinces * If [Address] != [Name] in [List of provinces], user can manually enter. | | Bio | * Rich text box * Length [Bio] >= 0 | |
| *(5)* | *BR3* | **Save changes rules:**  When user clicks on “Save changes” button,   * Data Fields are updated * The system will load MSG3 |

# Review app

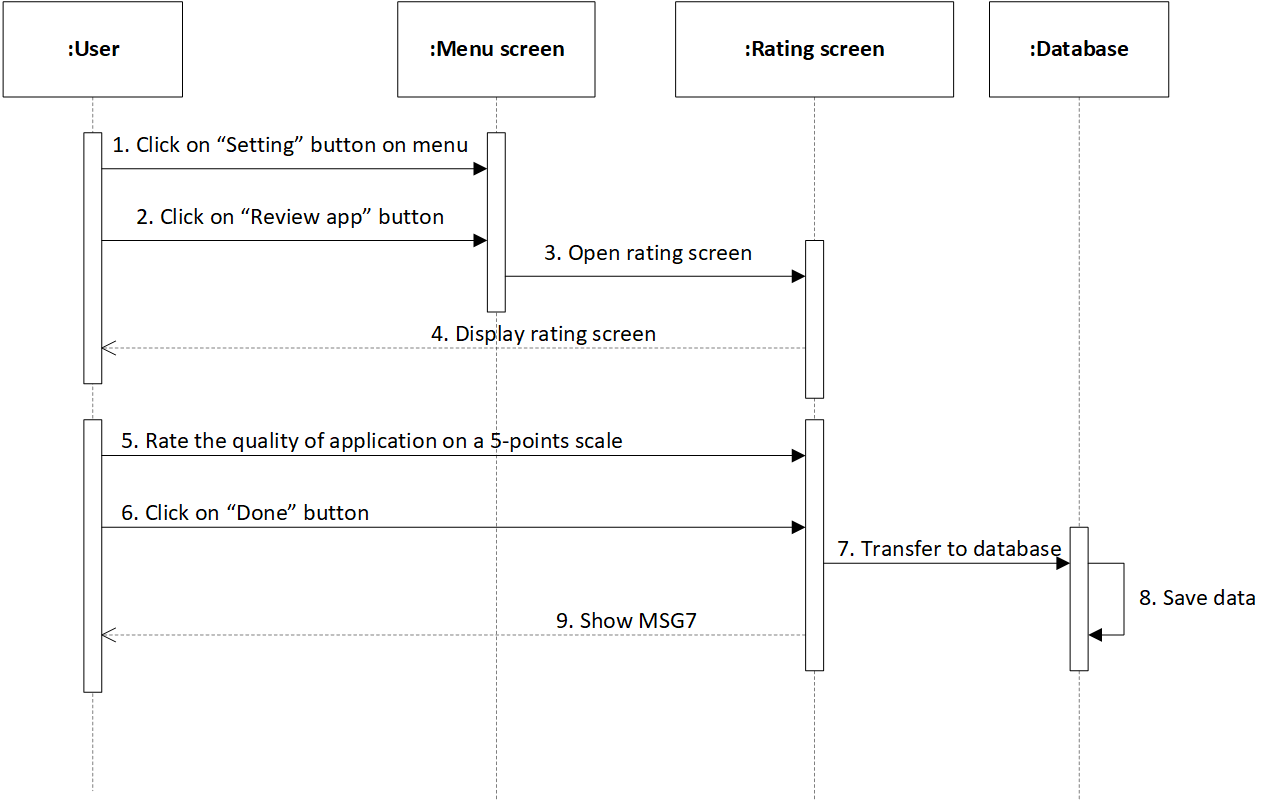
| **Name** | **Review app** |
| --- | --- |
| **Description** | This use case allows user to review app. |
| **Actor** | Normal User, Silver User, Gold User. |
| **Trigger** | * Open the application * Click on “Review app” button. |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Setting” button on menu * Click on “Review app” button |
| **Post-condition** | * Data is archived in database * Show MSG3 |

**Activity Flow**



*Figure 38: Activity Flow - Review app*

**Sequence Diagram**



*Figure 39: Sequence Diagram - Review app*

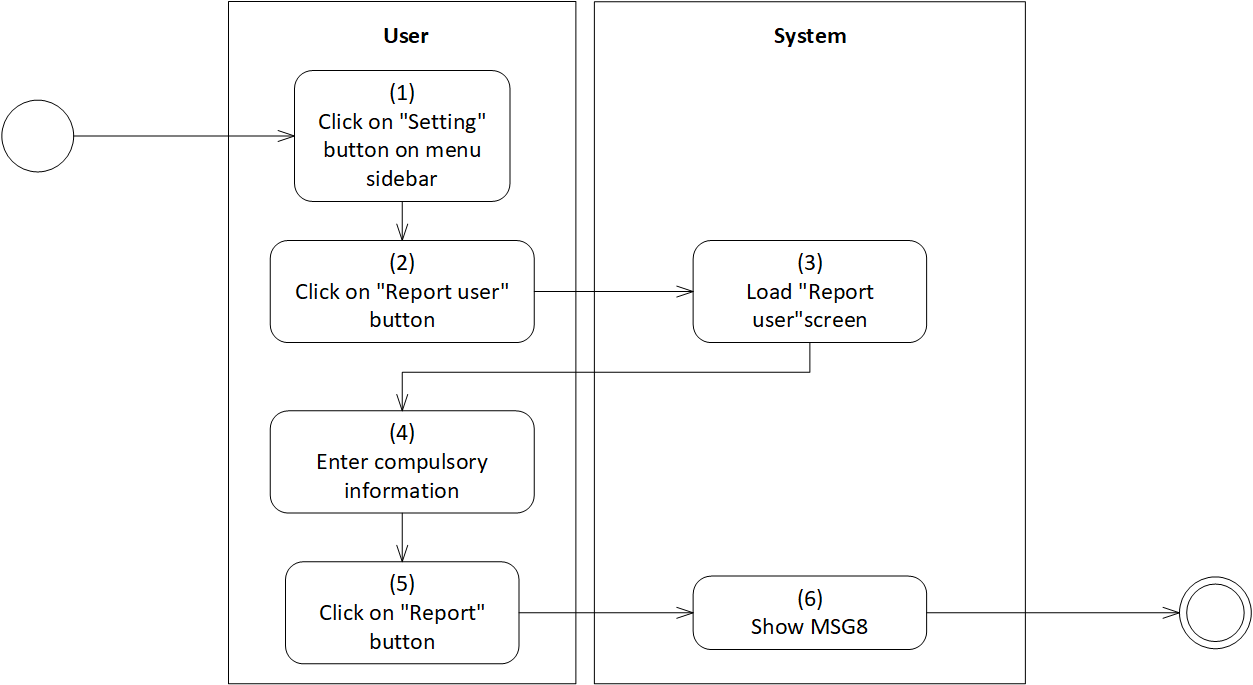
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “Rating” screen rules**:   * When user clicks on “Setting” button on menu sidebar, then clicks on “Review app” button, the system loads “Rating” screen. (Refer to “Rating” screen in “UI Description” file) |
| *(4)* | *BR2* | **Rating rules:**  User rate the quality of the application on the 5-point scale (from 1 to 5 points)  When user clicks on “Done” button, the system will load MSG7 |

# Report user

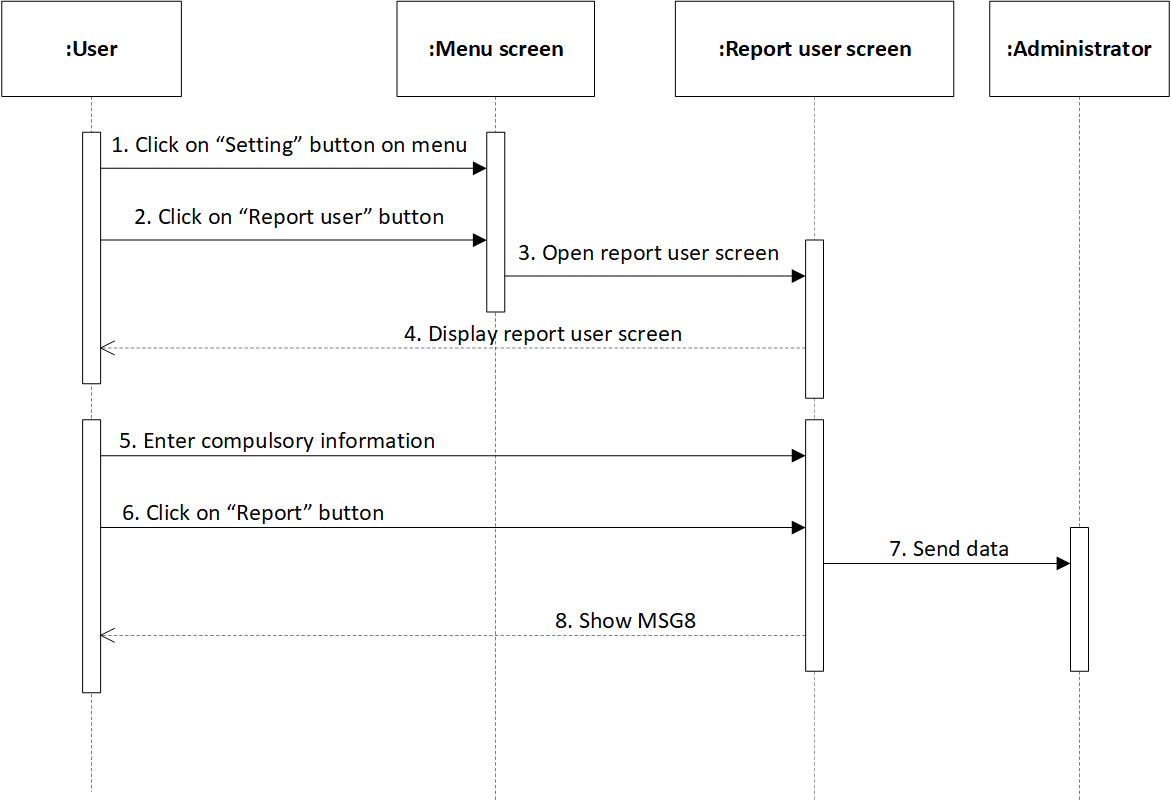
| **Name** | **Report user** |
| --- | --- |
| **Description** | This use case allows user to report other users who have non-standardized/ harassing behaviors. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Report user” button. |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Setting” button on menu * Click on “Report user” button |
| **Post-condition** | * Show MSG8 |

**Activity Flow**



*Figure 40: Activity Flow - Report user*

**Sequence Diagram**



*Figure 41: Sequence Diagram - Report user*

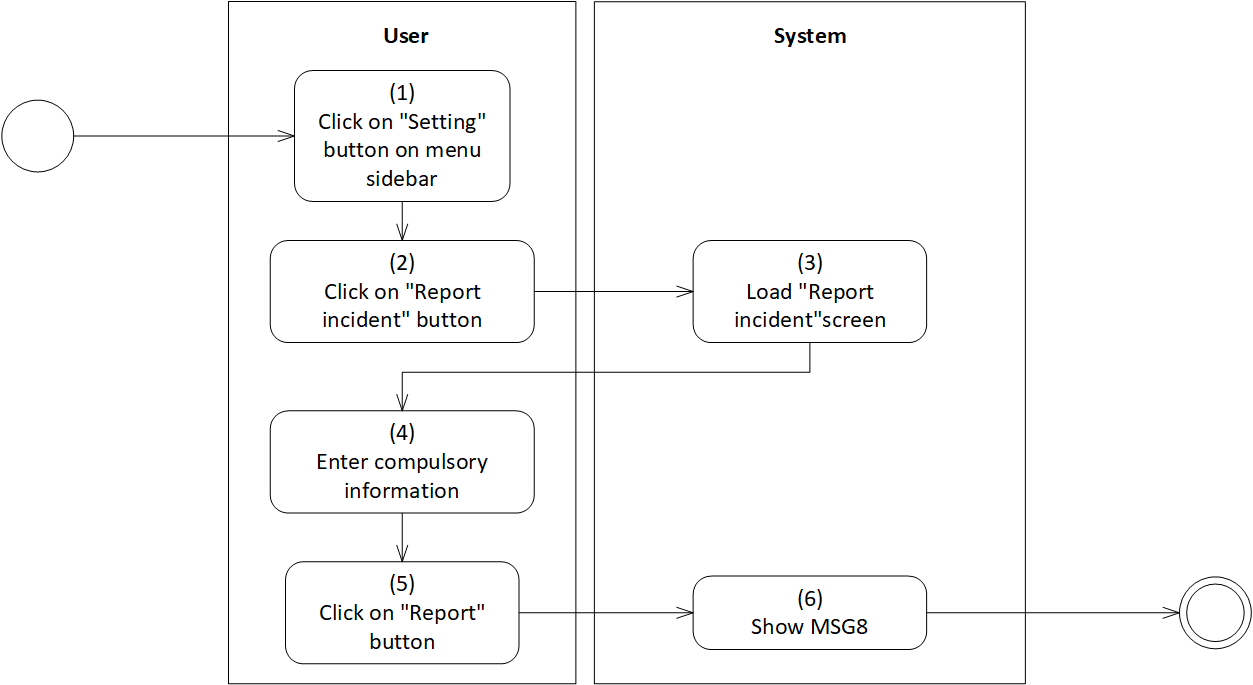
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “Report user” Form rules**:   * When user clicks on “Setting” button on menu sidebar, then clicks on “Report user” button, the system loads “Report user” form. (Refer to “Report user” list in “UI Description” file) |
| *(4)* | *BR2* | **Input checking rules:**  User enters compulsory information in the textboxes of “Report user” form. If user enters according to the accurate format, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” |
|  |  | **Report user Form**:  Fill information in “Report user” form as the template below:   | Reporter | * Get [Username] of current user | | --- | --- | | User reported | * Text box * Get [Username] of user reported | | Reason | * List of checkboxes demonstrate the reporting reasons * If all checkboxes have [isChecked] = False, user can not send a report. | | Other | * Rich text box * Length [Other] >= 0 | |
| *(5)* | *BR3* | **Report rules:**  When user clicks on “Report” button,   * If all checkboxes have [isChecked] = False, * The system will load MSG0 * User selects checkbox again * Else * Report is sent to administrator group * The system will load MSG8 |

# Report incident

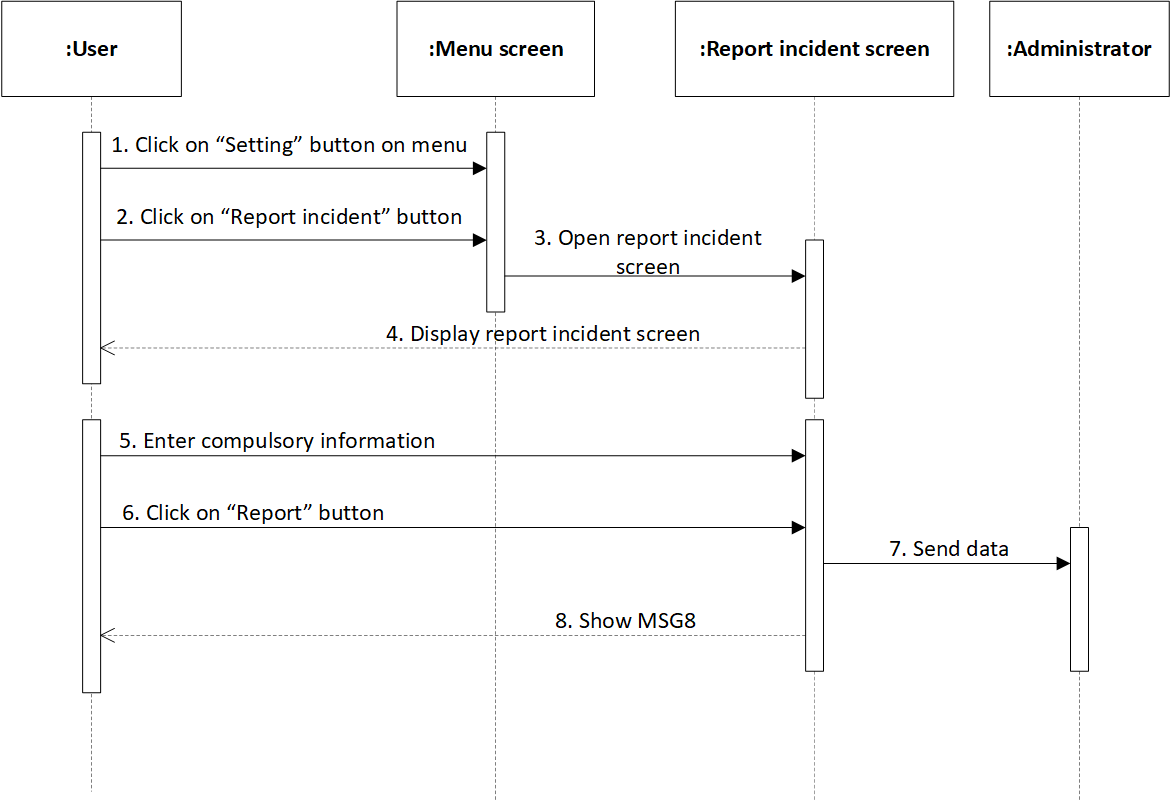
| **Name** | **Report incident** |
| --- | --- |
| **Description** | This use case allows users to report incidents that occurred during their usage time. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Open the application * Click on “Report incident” button. |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access the menu screen * Click on “Setting” button on menu * Click on “Report incident” button |
| **Post-condition** | * Show MSG8 |

**Activity Flow**



*Figure 42: Activity Flow - Report incident*

**Sequence Diagram**



*Figure 43: Sequence Diagram - Report incident*

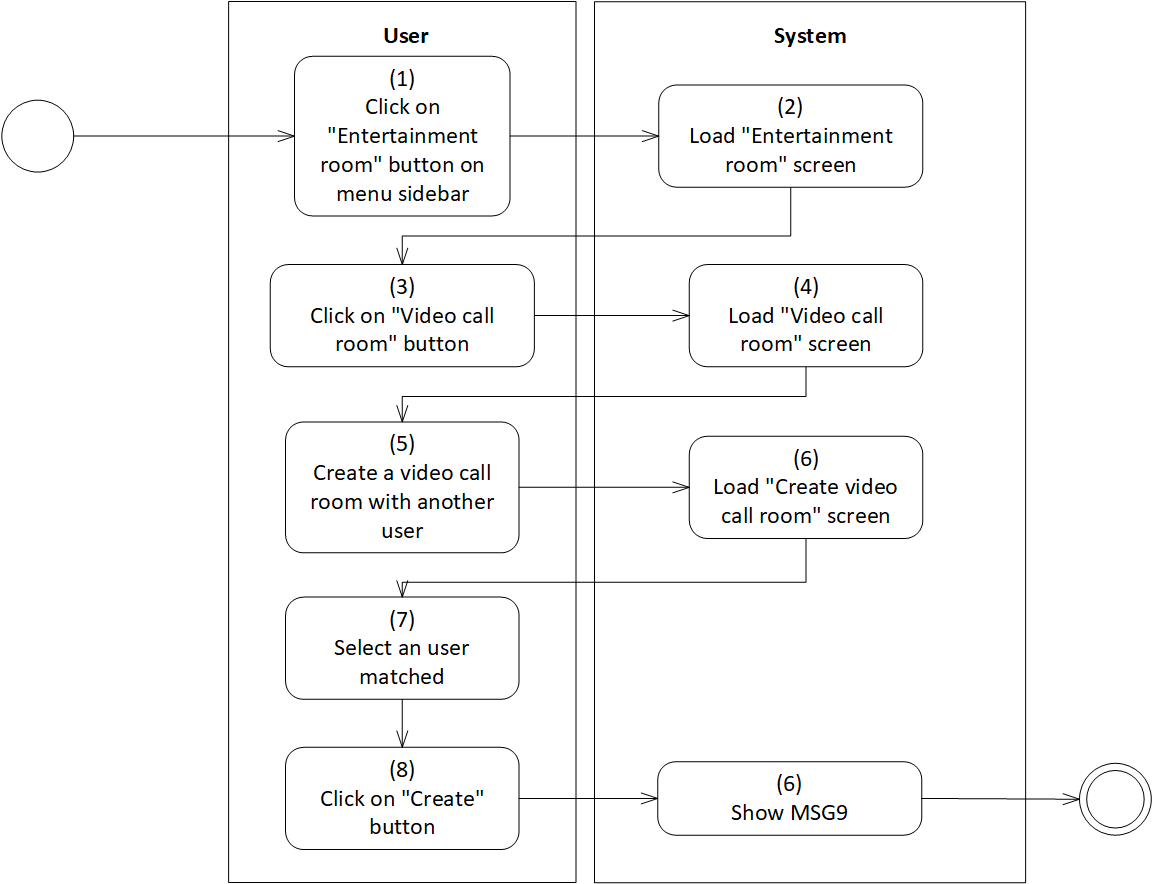
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | Load “Report incident” Form rules:   * When user clicks on “Setting” button on menu sidebar, then clicks on “Report incident” button, the system loads “Report incident” form. (Refer to “Report incident” screen in “UI Description” file) |
| *(4)* | *BR2* | **Input checking rules:**  User enters compulsory information in the textboxes of “Report incident” form. If user enters according to the accurate format, the system does nothing; else, the system will prompt an error message. The system will check user inputs as the following:   * If [Textbox Text] = “”, set [Text Error] = “<<Field Name>>” + “is a compulsory data. Please enter this data!” |
|  |  | **Report incident Form**:  Fill information in “Report user” form as the template below:   | Reporter | * Get [Username] of current user | | --- | --- | | Description | * Rich text box * Length [Description] >= 20 characters | |
| *(5)* | *BR3* | **Report rules:**  When user clicks on “Report” button,   * If [Description] = “”, * The system will load MSG0 * User demonstrates incident in [Description] * Else * Report is sent to administrator group * The system will load MSG8 |

# Create video call room

| **Name** | **Create video call room** |
| --- | --- |
| **Description** | This use case allows user to create a video call room with another user in the system. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Entertainment room” button on menu * Click on “Video call room” button * Click on “Create room” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Entertainment room” button * Click on “Video call room” button * Click on “Create room” button |
| **Post-condition** | * Create video call room successfully * Show MSG9. |

**Activity Flow**



*Figure 44: Activity Flow - Create video call room*

**Sequence Diagram**



*Figure 45: Sequence Diagram - Create video call room*

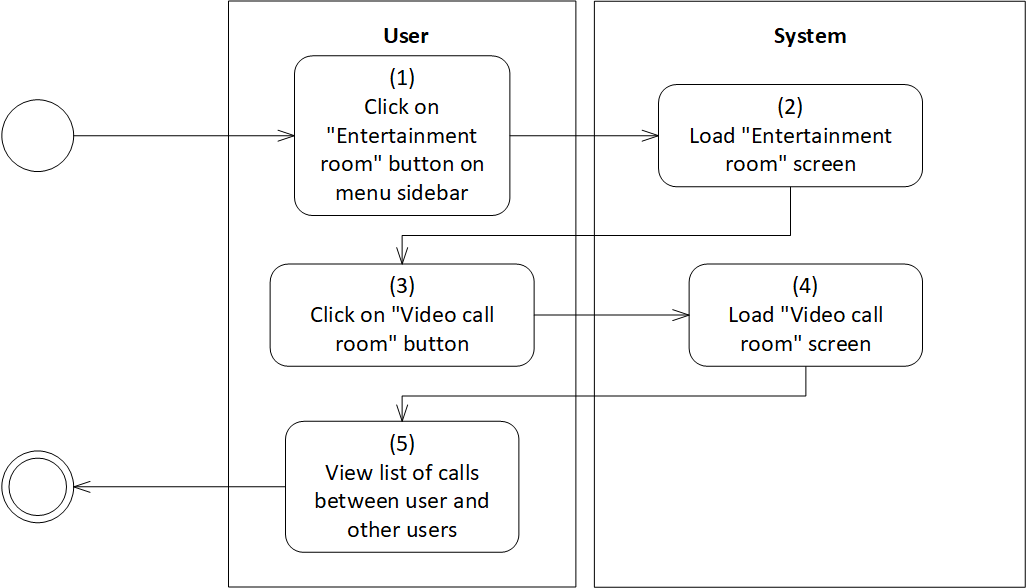
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Entertainment room” screen rules**:  When user clicks on “Entertainment room” button, the system loads “Entertainment room” screen. (Refer to “Entertainment room” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Video call room” screen rules**:  When user clicks on “Video call room” button, the system loads “Video call room” screen. (Refer to “Video call room” screen in “UI Description” file) |
| *(6)* | *BR3* | **Load “Create video call room” screen rules**:  When user clicks on “Create video call room” button, the system loads “Create video call room” screen. (Refer to “Create video call room” screen in “UI Description” file) |
| *(6)* | *BR3* | **Create video call room rules:**  User only select one user in the matching list.   * If [Matching List] = “”, user cannot create video call room * After user selects one user in the matching list, clicks on “Create” button. Then, the system will load MSG9 |
|  |  | **Video call rules:**  Each type of account will have different rules for chatting duration.   * If [Account Type] = “Silver”, set [Duration] = 10 minutes/user * If [Account Type] = “Gold”, set [Duration] = 30 minutes/user |

# View calling history

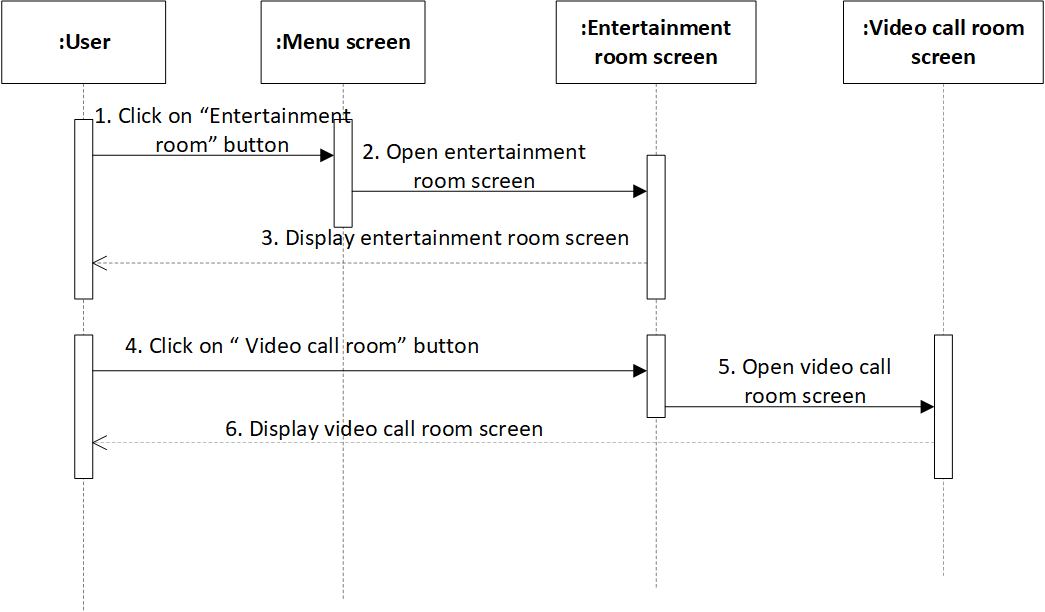
| **Name** | **View calling history** |
| --- | --- |
| **Description** | This use case allows user to review list of calls. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Entertainment room” button on menu * Click on “Video call room” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Entertainment room” button * Click on “Video call room” button |
| **Post-condition** | * Review list of calls. |

**Activity Flow**



*Figure 46: Activity Flow - View calling history*

**Sequence Diagram**



*Figure 47: Sequence Diagram - View calling history*

**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Entertainment room” screen rules**:  When user clicks on “Entertainment room” button, the system loads “Entertainment room” screen. (Refer to “Entertainment room” list in “List Description” file) |
| *(4)* | *BR2* | **Load “Video call room” screen rules**:  When user clicks on “Video call room” button, the system loads “Video call room” screen. (Refer to “Video call room” list in “List Description” file) |
| *(5)* | *BR3* | **List of calls rules:**  In the list calls, each call will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | | Duration | Get [Duration] of the call | |

# View matching history

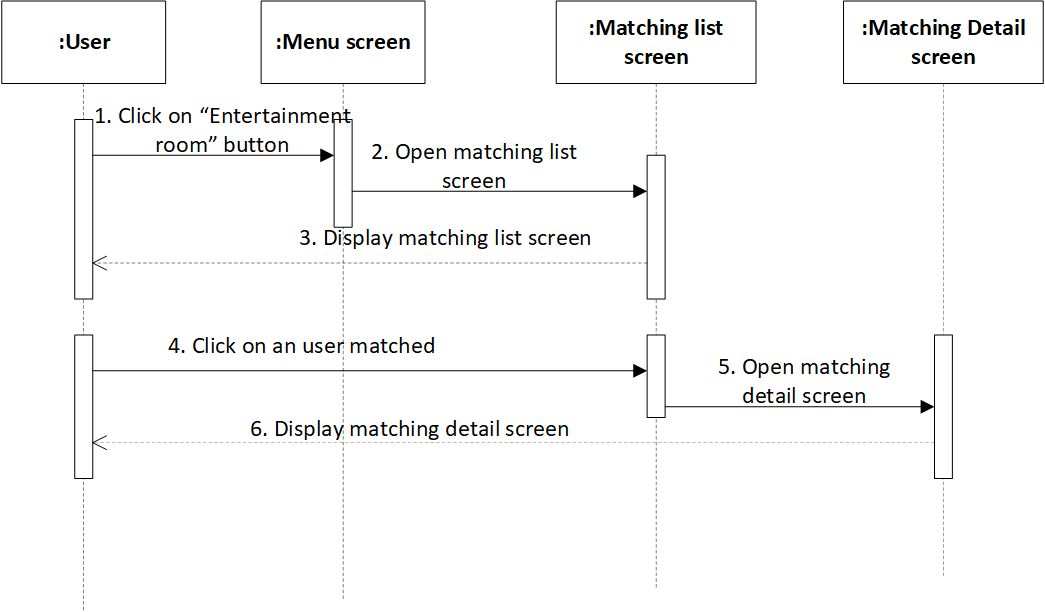
| **Name** | **View matching history** |
| --- | --- |
| **Description** | This use case allows user to review list of users matched. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Matching List” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Matching List” button |
| **Post-condition** | * View list of matching times. |

**Activity Flow**



*Figure 48: Activity Flow - View matching history*

**Sequence Diagram**



*Figure 49: Sequence Diagram - View matching history*

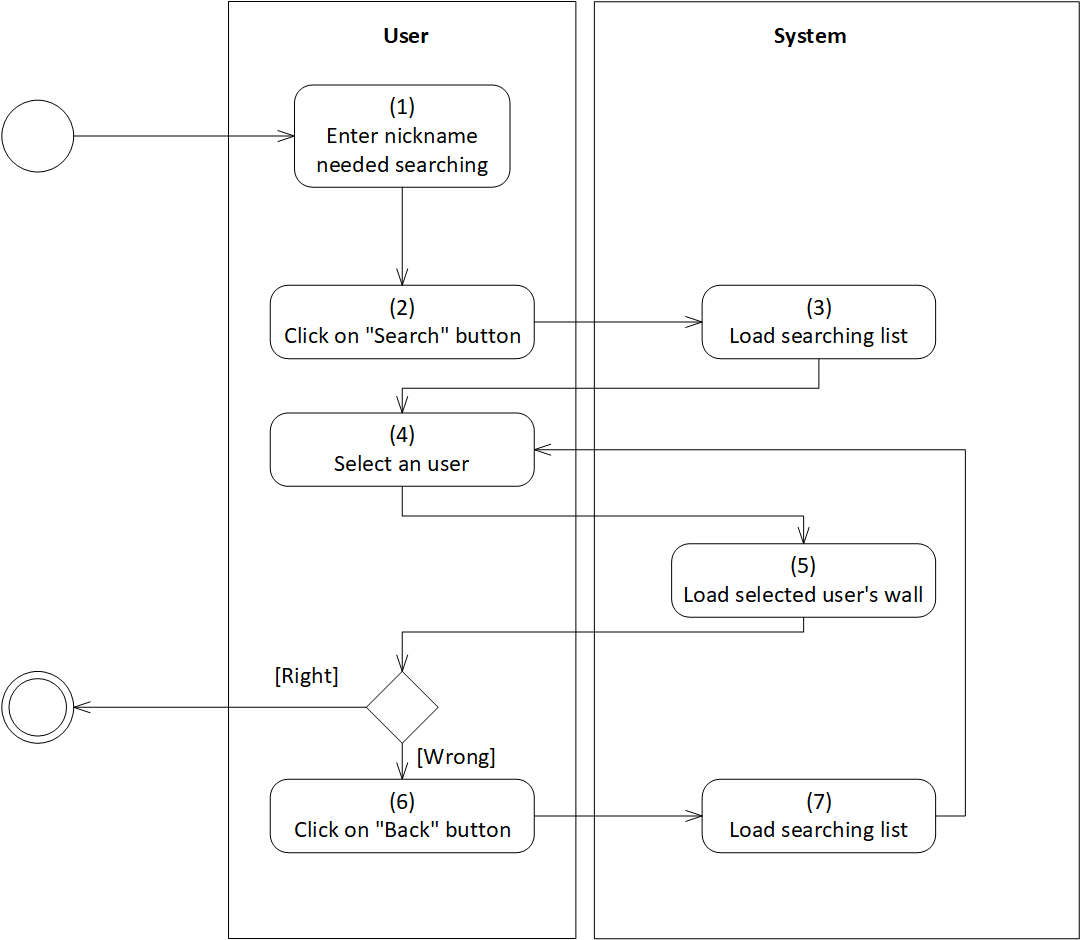
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Matching List” screen rules**:  When user clicks on “Matching List” button, the system loads “Matching List” screen. (Refer to “Matching List” screen in “UI Description” file) |
| *(3)* | *BR2* | **List of matched users:**  The list of matched users will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | | Matching Time | Get [Matching Time] | |
| *(5)* | *BR3* | **Load “Matching Details” Form rules**:  When user clicks on each user, the system loads “Matching Details” form. (Refer to “Matching Details” list in “UI Description” file) |
| *(6)* | *BR4* | **View details rules:**  User can click on a card representing each user to view details. The matching details will include:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Avatar | Get [Avatar] of current user | | Username | Get [Username] of selected user | | Username | Get [Username] of current user | | Hobbies | Get [Hobbies] of selected user where [Hobbies] = [Hobbies] of current user | | Body | [Body] = “You and” + [Username] of selected user + “matched successfully!” + “on” + [Matching Time]  [Body] = [Body] + new line  [Body] = [Body] + “Both of you have hobbies” + [Hobbies] | | Matching Time | Get [Matching Time] of the nearest message | |

# Search user

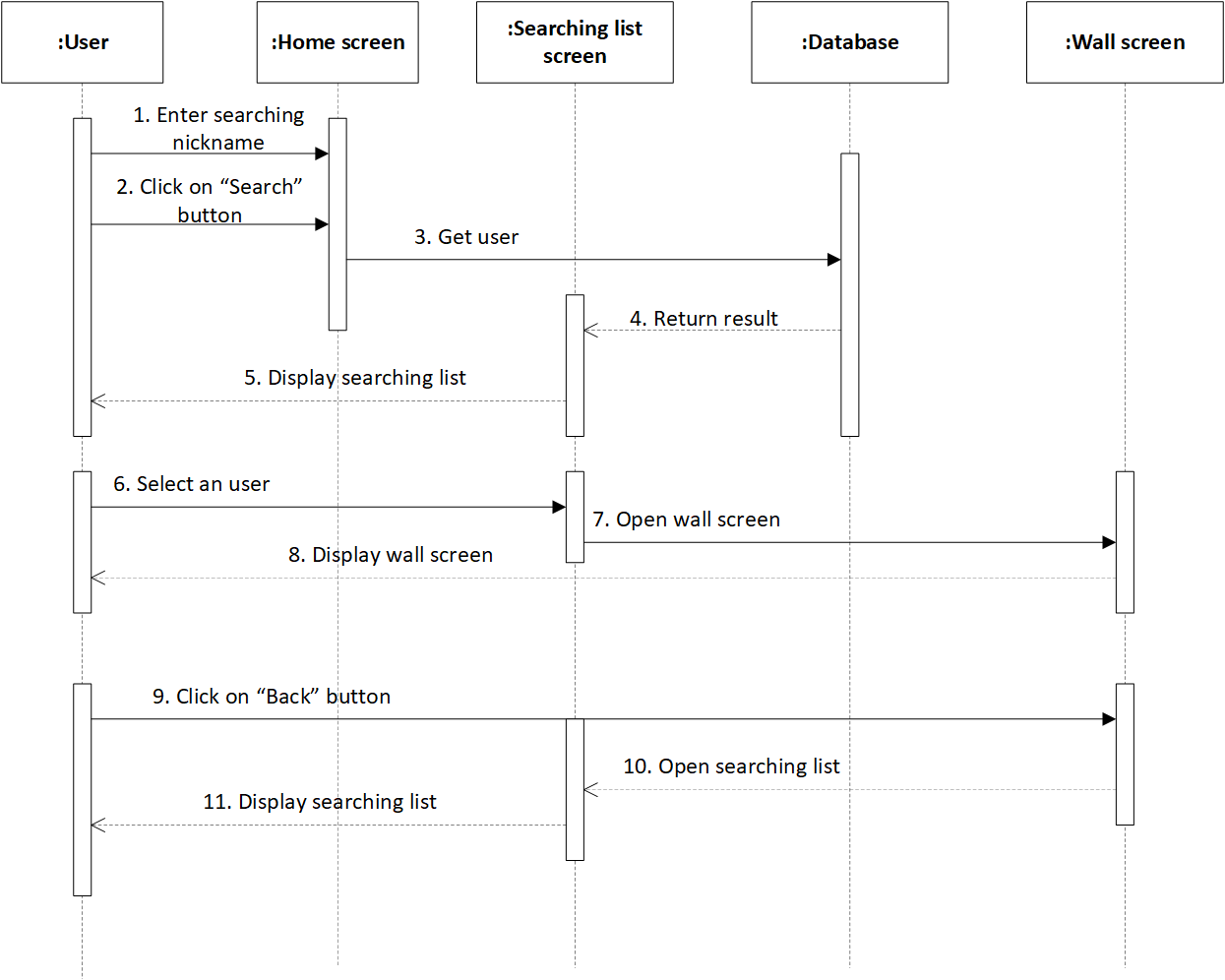
| **Name** | **Search user** |
| --- | --- |
| **Description** | This use case allows user to search other users based on user nickname. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Search” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access “Home” screen * Click on “Search” button |
| **Post-condition** | * Find out desired users. |

**Activity Flow**



*Figure 50: Activity Flow - Search user*

**Sequence Diagram**



*Figure 51: Sequence Diagram - Search user*

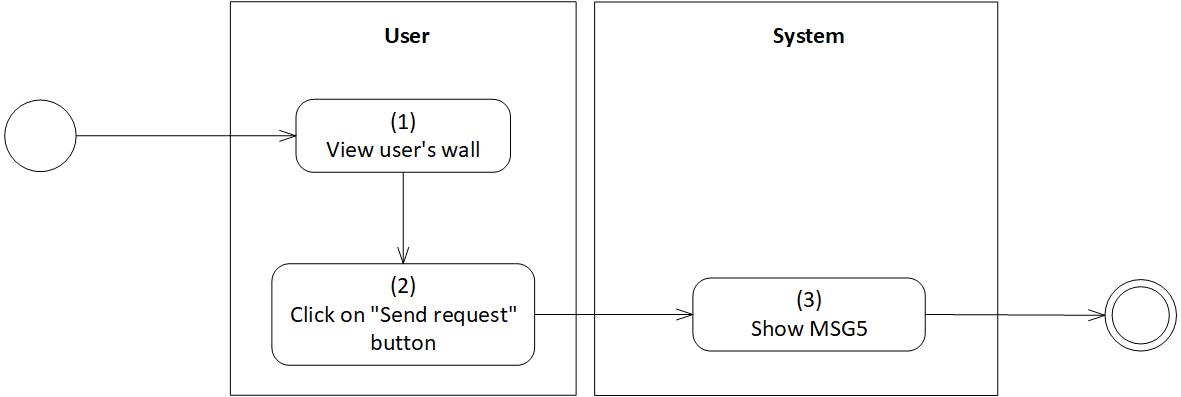
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Input rules**:  When user enter searching nickname in the textbox, the system will check each character to find out the suitable users as the following:   * For [char] in [Textbox Text], the system retrieves [Username] so that [char] in [Username] |
| *(3)* | *BR2* | **Load searching list rules**:  When user clicks on “Search” button on menu sidebar, the system loads searching list. |
| *(5)* | *BR3* | **Load “Wall” screen rules**:  When user clicks on each user in searching list, the system loads “Wall” screen (Refer to “Wall” screen in “UI Description” file) |

# Match with searching user

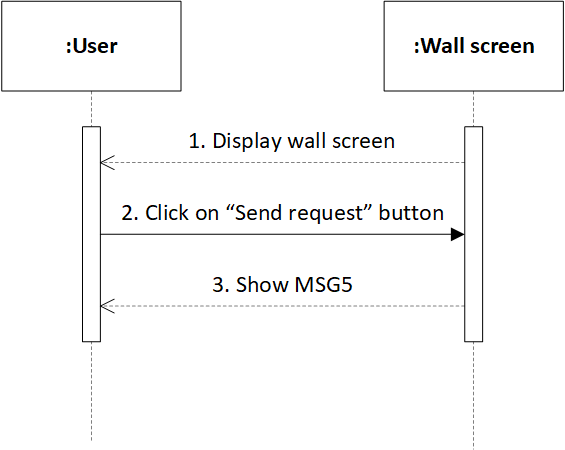
| **Name** | **Match with searching user** |
| --- | --- |
| **Description** | This use case allows user to send matching request with user searched |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Send request” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access searching user’s wall * Click on “Send request” button |
| **Post-condition** | * Send matching request successfully. |

**Activity Flow**



*Figure 52: Activity Flow - Match with searching user*

**Sequence Diagram**



*Figure 53: Sequence Diagram - Match with searching user*

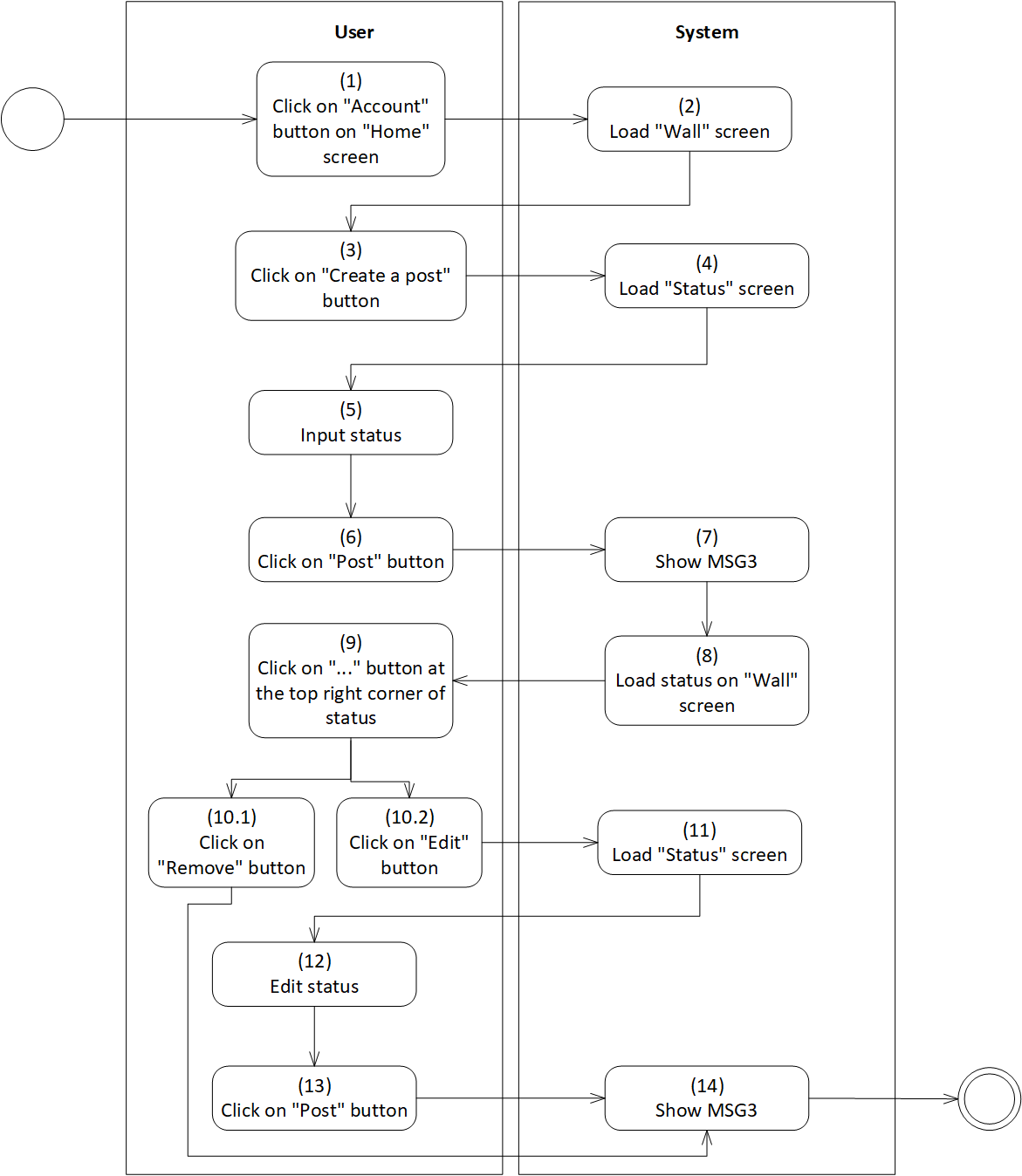
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Load “Wall” screen rules**:  When user clicks on each user in searching list, the system loads “Wall” screen (Refer to “Wall” screen in “UI Description” file) |
| *(2)* | *BR2* | **Send request rules**:  User clicks on “Send request” button of the “Wall” screen, the system will load MSG5 |

# CRUD Status

| **Name** | **CRUD Status** |
| --- | --- |
| **Description** | This use case allows user to add, remove, edit status posted on the wall |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Click on “Add” button * Click on “Remove” button * Click on “Edit” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access wall * Click on “Add” button if [Demand]== “Add” * Otherwise, click on “Remove” button if [Demand]== “Remove” * Otherwise, click on “Edit” button if [Demand]== “Edit” |
| **Post-condition** | * The status is added/removed/edited. * Show MSG3 |

**Activity Flow**



*Figure 54: Activity Flow - CRUD Status*

**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Wall” screen rules**:  When user clicks on “Account” button on “Home” screen, the system loads “Wall” screen (Refer to “Wall” screen in “UI Description” file) |
| *(4)* | *BR2* | **Load “Status” Form rules**:  When user clicks on “Create a post” button on “Home” screen, the system loads “Status” form (Refer to “Status” list in “UI Description” file) |
| *(5)* | *BR3* | **Input checking rules**:  User enters the status in the rich text box. The system will check input as the following:   * If [RichTextBox Text] = “”, set button “Post” = Disabled * Else, set button “Post” = Enabled |
| *(10.1)* | *BR4* | **Remove status rules**:  After user clicks on "..." button at the top right corner of status, clicks on “Remove” button.   * If the system processes successfully, loads MSG3 * Else, loads MSG0 |
| *(10.2)* | *BR5* | **Edit status rules**:  User clicks on "..." button at the top right corner of status.  When user continue clicking on “Edit” button, the system loads “Status” screen |
|  |  | **Input checking rules**:  User enters the status in the rich text box. The system will check input as the following:   * If [RichTextBox Text] = “”, set button “Post” = Disabled * Else, set button “Post” = Enabled |

# 

# React to user post

| **Name** | **React to user post** |
| --- | --- |
| **Description** | This use case allows user to react posts posted by themselves and other users. |
| **Actor** | Silver User, Gold User |
| **Trigger** | * Have signed up already |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already |
| **Post-condition** | * The status is added/removed/edited. * Show MSG3 |

**Activity Flow**

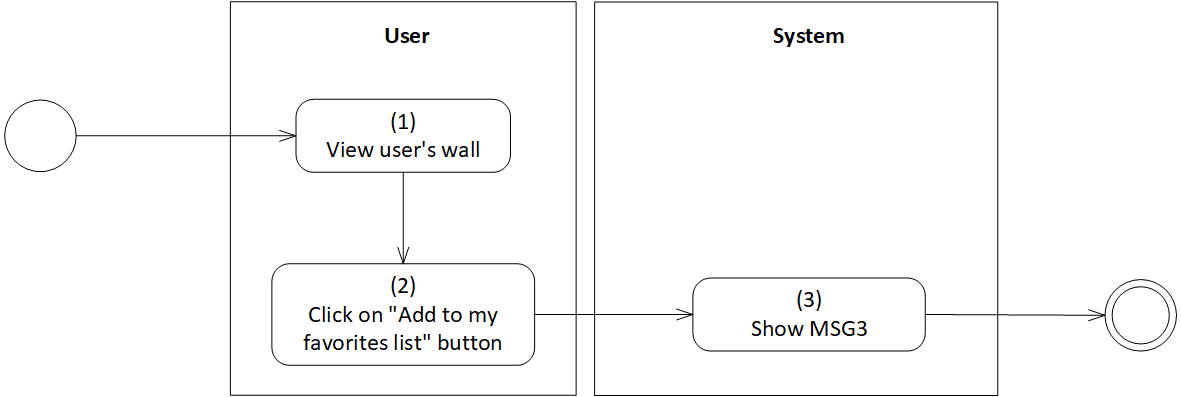


*Figure 55: Activity Flow - React to user post*

# Add to the Favorites list

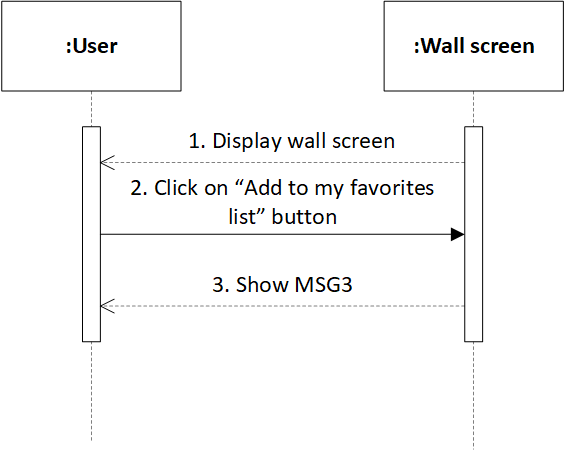
| **Name** | **Add to the Favourites list** |
| --- | --- |
| **Description** | This use case allows user to add users to the Favorites list. |
| **Actor** | Gold User |
| **Trigger** | * Click on “Add to my favorites list” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access user’s wall * Click on “Add to my favorites list” button |
| **Post-condition** | * Other users are added to favorites list. * Show MSG3 |

**Activity Flow**



*Figure 56: Activity Flow - Add to the favorites list*

**Sequence Diagram**



*Figure 57: Sequence Diagram - Add to the favorites list*

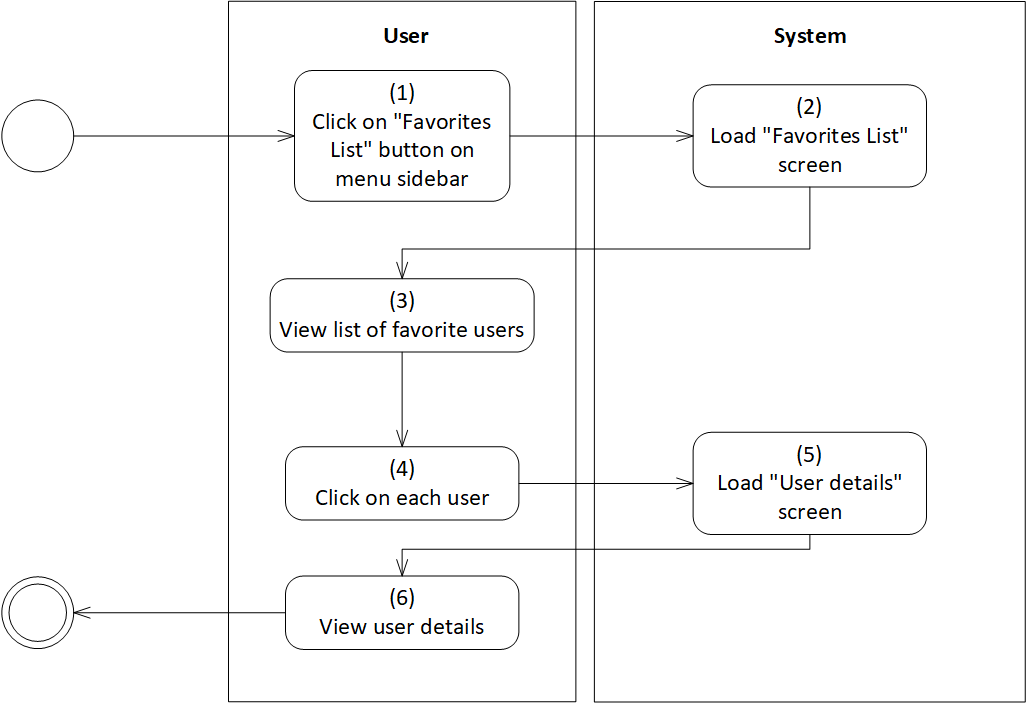
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Load “Wall” screen rules**:  When user clicks on each user in searching list, the system loads “Wall” screen (Refer to “Wall” screen in “UI Description” file) |
| *(2)* | *BR2* | **Add to favourites list rules**:  User clicks on “Add to my favourites list” button of the “Wall” screen, the system will load MSG3 |

# View favorites list

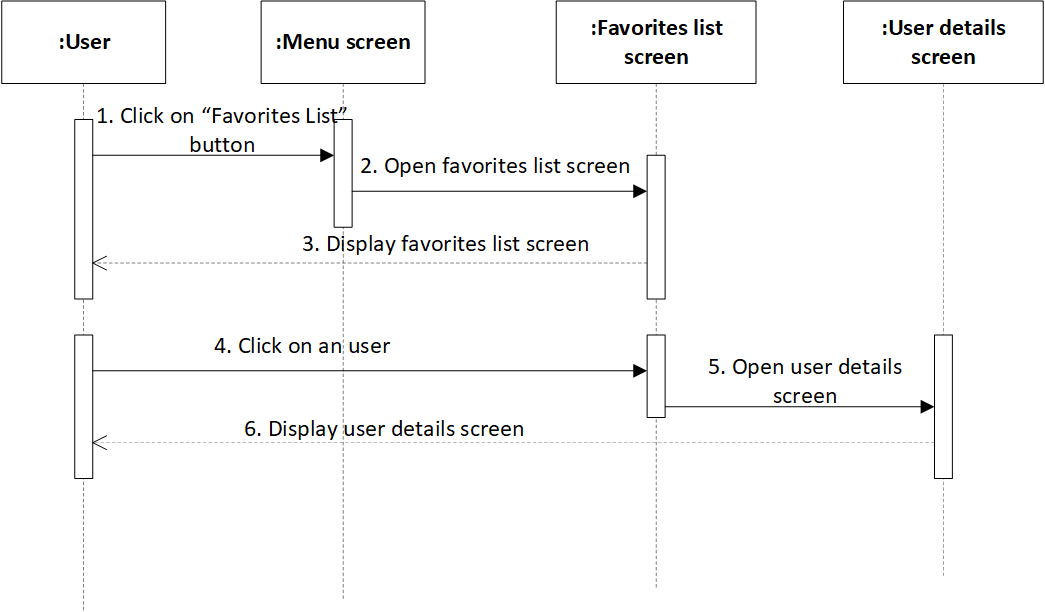
| **Name** | **View favourites list** |
| --- | --- |
| **Description** | This use case allows user to view list of favorite users. |
| **Actor** | Gold User |
| **Trigger** | * Click on “Favorites List” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Favorites List” button |
| **Post-condition** | * View favorite users |

**Activity Flow**



*Figure 58: Activity Flow - View Favorites list*

**Sequence Diagram**



*Figure 59: Sequence Diagram - View Favorites list*

**Business Rules**

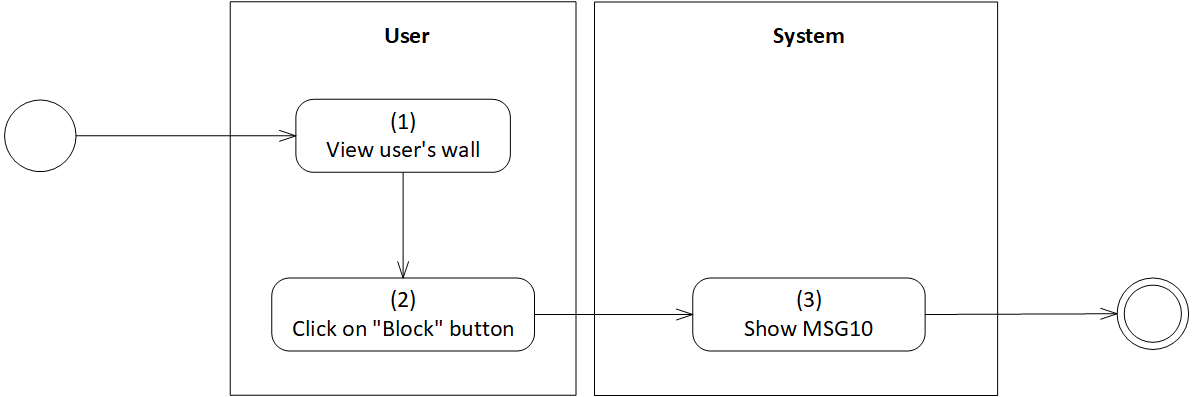
| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “Favourites List” screen rules**:  When user clicks on “Favourites List” button, the system loads “Favourites List” screen. (Refer to “Favourites List” screen in “UI Description” file) |
| *(3)* | *BR2* | **List of favourite users:**  The list of favourite users will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | |
| *(5)* | *BR3* | **Load “User Details” Form rules**:  When user clicks on each user, the system loads “User Details” form. (Refer to “User Details” list in “UI Description” file) |
| *(6)* | *BR4* | **View details rules:**  User can click on a card represents for each user to view details. The user details will include:   | Gender | Get [Gender] of selected user | | --- | --- | | Age | Get [Age] of selected user | | Distance | Get [Distance] between selected user and current user | | Education | Get [Education] of selected user | | Profession | Get [Profession] of selected user | | Hobbies | Get [Hobbies] of selected user | |

# 

# Block user

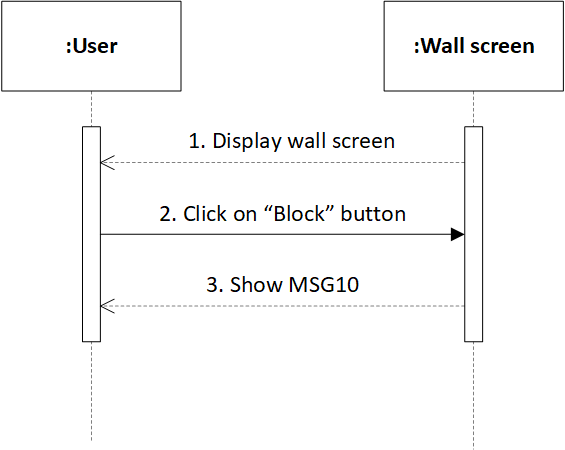
| **Name** | **Block user** |
| --- | --- |
| **Description** | This use case allows user to block other user accounts. |
| **Actor** | Gold User |
| **Trigger** | * Click on “Block” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access user’s wall * Click on “Block” button |
| **Post-condition** | * Account is blocked. * Show MSG10 |

**Activity Flow**



*Figure 60: Activity Flow - Block user*

**Sequence Diagram**



*Figure 61: Sequence Diagram - Block user*

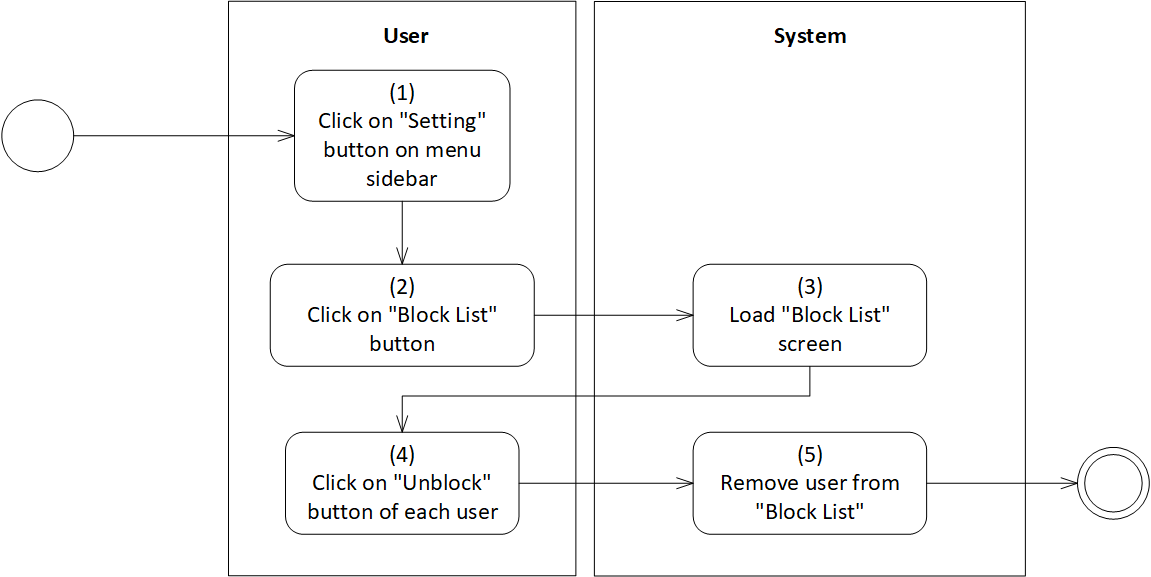
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Load “Wall” screen rules**:  When user clicks on each user in searching list, the system loads “Wall” screen (Refer to “Wall” list in “List Description” file) |
| *(2)* | *BR2* | **Block user account rules**:  User clicks on “Block” button of the “Wall” screen, the system will load MSG10 |

# Unblock user

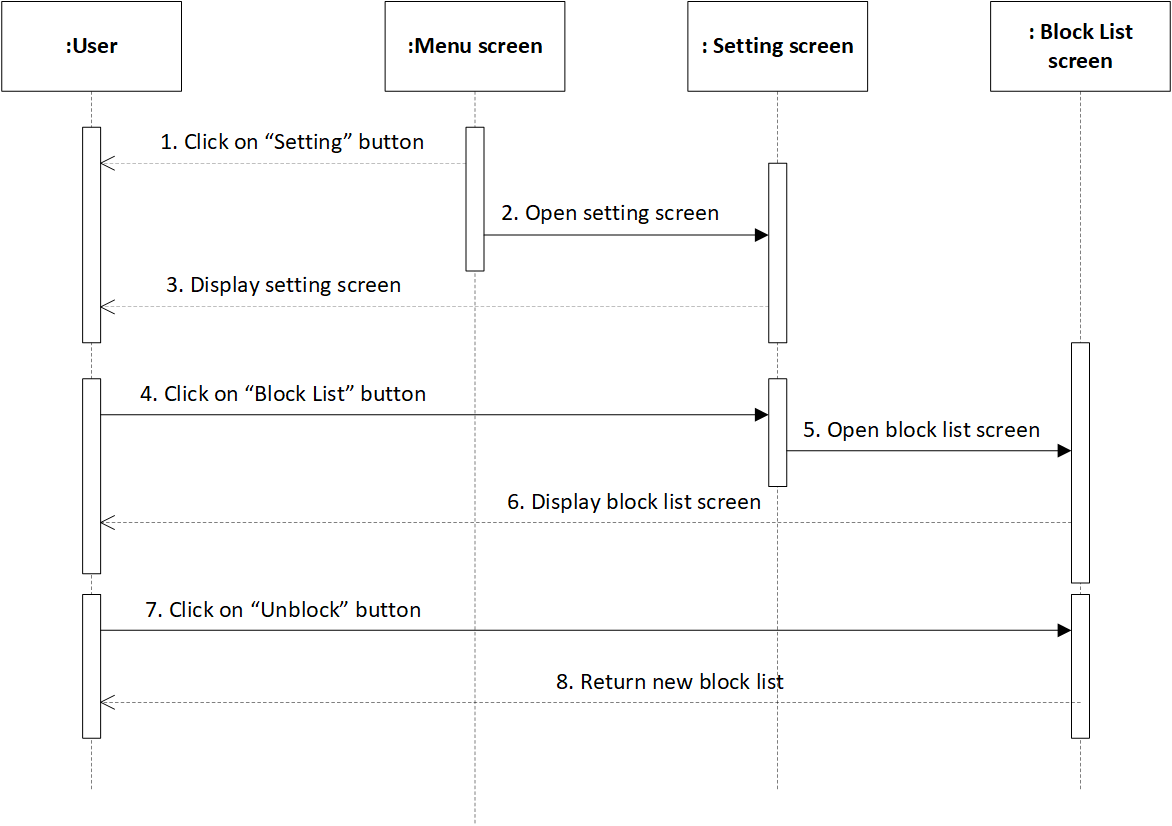
| **Name** | **Unblock user** |
| --- | --- |
| **Description** | This use case allows user to unblock other user accounts. |
| **Actor** | Gold User |
| **Trigger** | * Click on “Unblock” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “Setting” button on menu * Click on “Block List” button |
| **Post-condition** | * Unblock account successfully. |

**Activity Flow**



*Figure 62: Activity Flow - Unblock user*

**Sequence Diagram**



*Figure 63: Sequence Diagram - Unblock user*

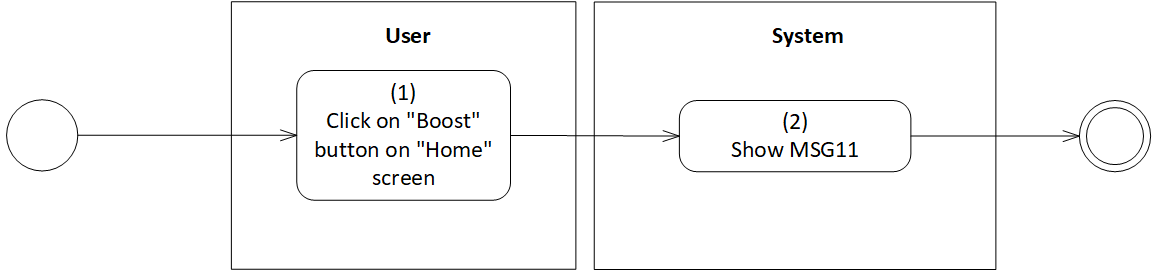
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “Block List” screen rules**:  When user clicks on “Setting” button, then clicks on “Block List” button, the system loads “Block List” screen. (Refer to “Block List” screen in “UI Description” file) |
| *(4)* | *BR2* | **List of blocked users:**  The list of matched users will be displayed as the template blow:   | Avatar | Get [Avatar] of selected user | | --- | --- | | Username | Get [Username] of selected user | |
|  |  | **Unblock rules:**  When user clicks on “Unblock” button on the right of a card that represents each user, the system will remove this card from the “Block List”. The system responses as the following:   * Assume [Block List] = [User Card 1] + [User Card 2] + … + [User Card n-1] + [User Card n] * With i in range 1 -> n * If [User Card i] has event [OnClicked] = True, set [Block List] = [Block List] – [User Card i] * Else, [Block List] is kept intact |

# Boost account

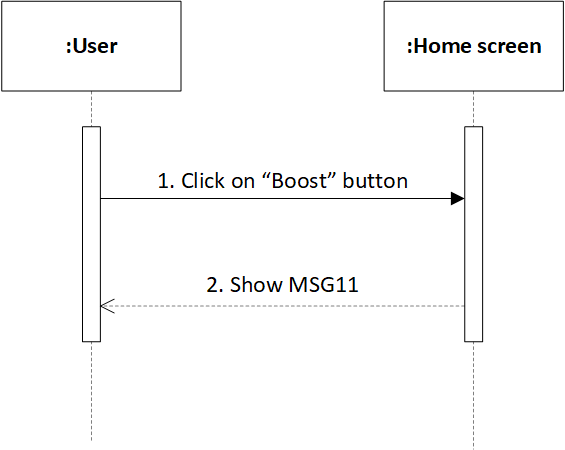
| **Name** | **Boost account** |
| --- | --- |
| **Description** | This use case allows user to make user profile become one of the top ones in user’s area for 30 minutes. |
| **Actor** | Gold User |
| **Trigger** | * Click on “Boost” button |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access home screen * Click on “Boost” button |
| **Post-condition** | * Account is boosted on the top of recommended list. * Show MSG11 |

**Activity Flow**



*Figure 64: Activity Flow - Boost account*

**Sequence Diagram**



*Figure 65: Sequence Diagram - Boost account*

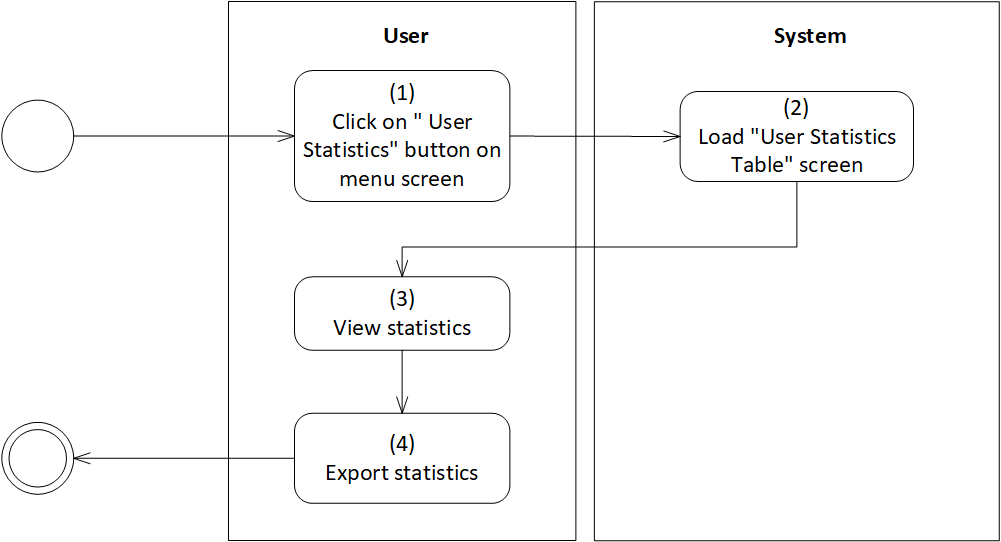
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(1)* | *BR1* | **Boost account rules**:  When user clicks on “Boost” button, this user account will be boosted to the top of their area in 30 minutes. |

# Create user statistics table

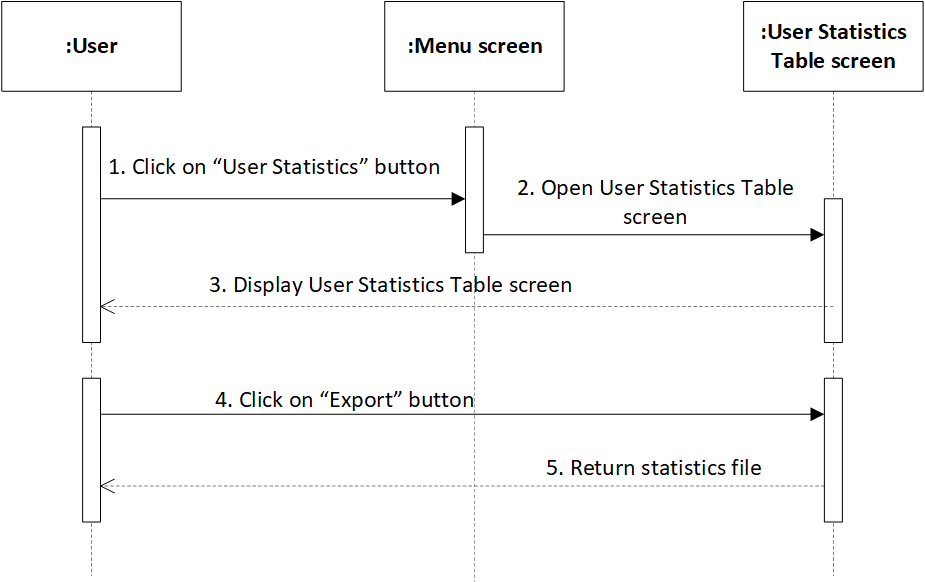
| **Name** | **Create user statistics table** |
| --- | --- |
| **Description** | This use case allows administrator to keep track of user statistics. |
| **Actor** | Administrator |
| **Trigger** | * Click on “User Statistics” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “User Statistics” button |
| **Post-condition** | * View user data statistics. * Export statistics |

**Activity Flow**



*Figure 66: Activity Flow - Create user statistics table*

**Sequence Diagram**



*Figure 67: Sequence Diagram - Create user statistics table*

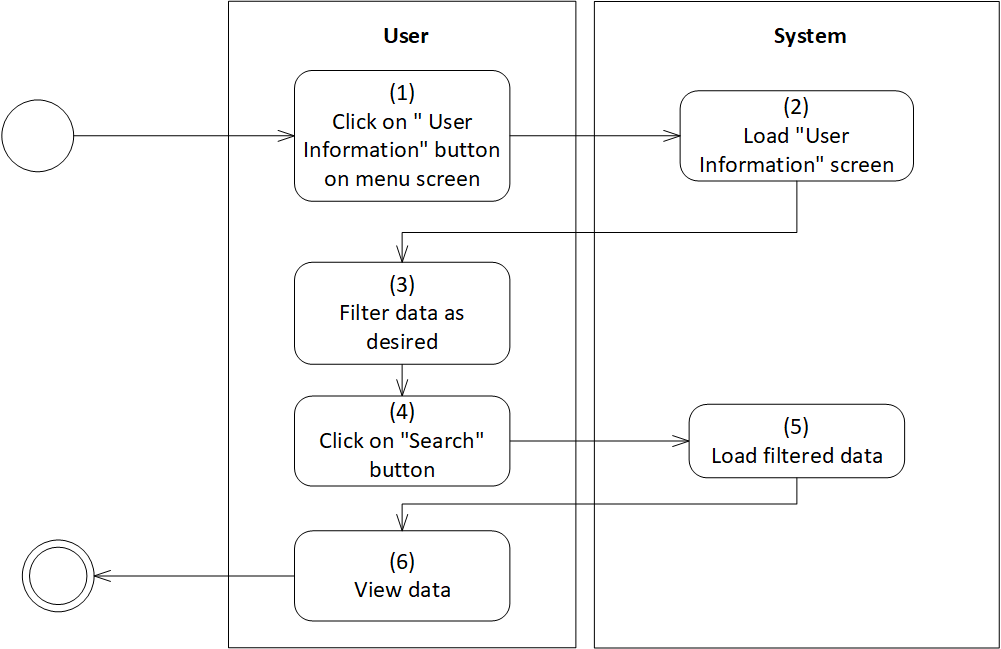
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Statistics Table” screen rules**:  When user clicks on “User Statistics” button, the system loads “User Statistics Table” screen. (Refer to “User Statistics Table” list in “List Description” file) |
| *(3)* | *BR2* | **User Statistics Table:**  The user statistics table will be displayed as the template blow:   | Total number of users | A card displays [Total number of user] | | --- | --- | | Number of users classified by gender | A bar chat with the x- axis is gender and the y- axis the number of each gender type | | Number of accounts classify by account type | A pie chart displays the proportion of account type | | Average active time | A card displays [Average Time] | | Top 10 provinces have the highest number of users | A table with the rows is the province names and the column is the number of users. Each cell will display the number of users belonging to each province. | | Top 10 provinces have the least number of users | A table with the rows is the province names and the column is the number of users. Each cell will display the number of users belongs to each province. | | Average rating point | A card displays [Rating Point] | |
| *(4)* | *BR3* | **Export statistics rules:**  When user clicks on “Export” button on “User Statistics Table” screen, the system will automatically create a file that contains current user statistics, and this file is downloaded to the user device. |

# View user information

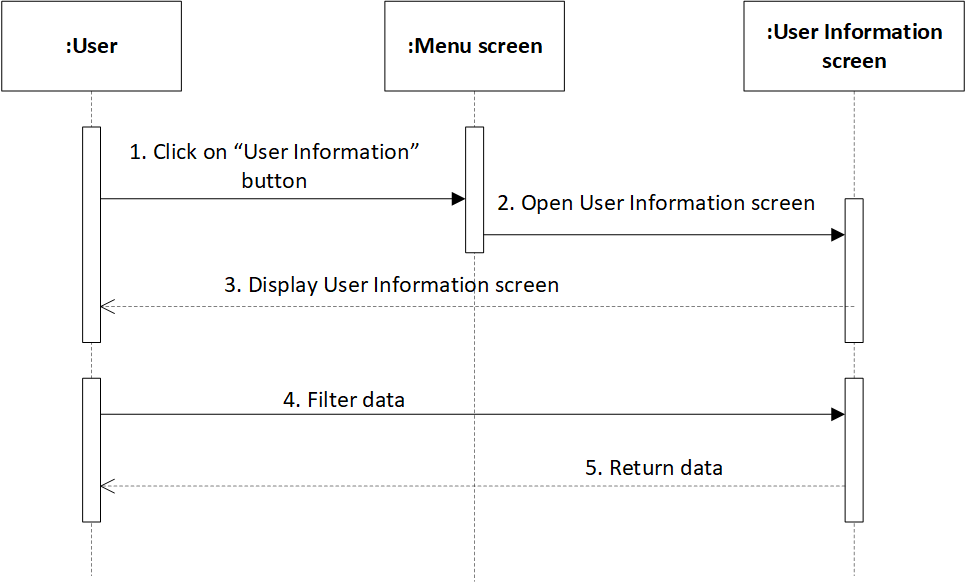
| **Name** | **View user information** |
| --- | --- |
| **Description** | This use case allows administrator to keep track of user information based on some specific fields. |
| **Actor** | Administrator |
| **Trigger** | * Click on “User Information” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “User Information” button |
| **Post-condition** | * View user information. |

**Activity Flow**



*Figure 68: Activity Flow - View user information*

**Sequence Diagram**



*Figure 69: Sequence Diagram - View user information*

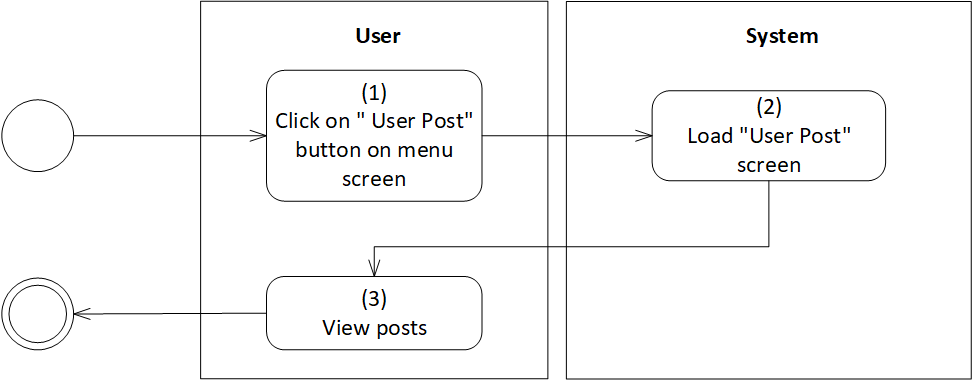
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Information” screen rules**:  When user clicks on “User Information” button, the system loads “User Information” screen. (Refer to “User Information” screen in “UI Description” file) |
| *(3)* | *BR2* | **Filter data rules**:  User can filter data according to some data field as below:   | Gender | [Gender] = {“Female”, “Male”, “LGBT”} | | --- | --- | | Region | [Region] = {“North”, “Central”, “South”} | | Education | [Education] = {“Secondary school”, “High school”, “College”, “University”, “Work”} | | Profession | [Profession] = {“Education”, “Entertainment”, “Business”, “Sport”, “Tech”, “Politics”} | | Age | [Age] = {(18,25), (26, 35), (36, 42), (43,50)} | |
| *(6)* | *BR3* | **View data rules:**  When user clicks on “Search” button, the system will load filtered data. Data is display as the template below:   | Username | Get [Username] of user | | --- | --- | | Age | Get [Age] of user | | Gender | Get [Gender] of user | | Education | Get [Education] of user | | Province | Get [Province] of user | | Region | Get [Region] of user | | Profession | Get [Profession] of user | |

# View users’ posts

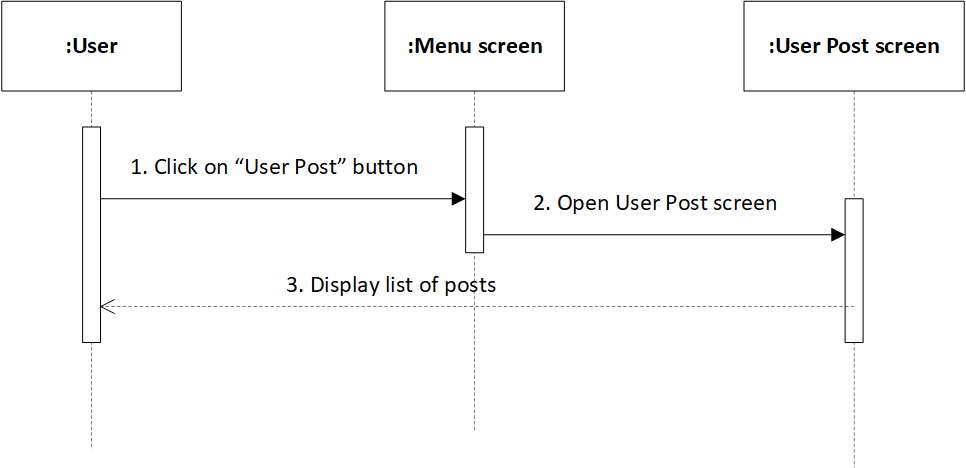
| **Name** | **View users’ posts** |
| --- | --- |
| **Description** | This use case allows administrator to view list of users’ posts. |
| **Actor** | Administrator |
| **Trigger** | * Click on “User Post” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “User Post” button |
| **Post-condition** | * View users’ posts. |

**Activity Flow**



*Figure 70: Activity Flow - View user posts*

**Sequence Diagram**



*Figure 71: Sequence Diagram - View user posts*

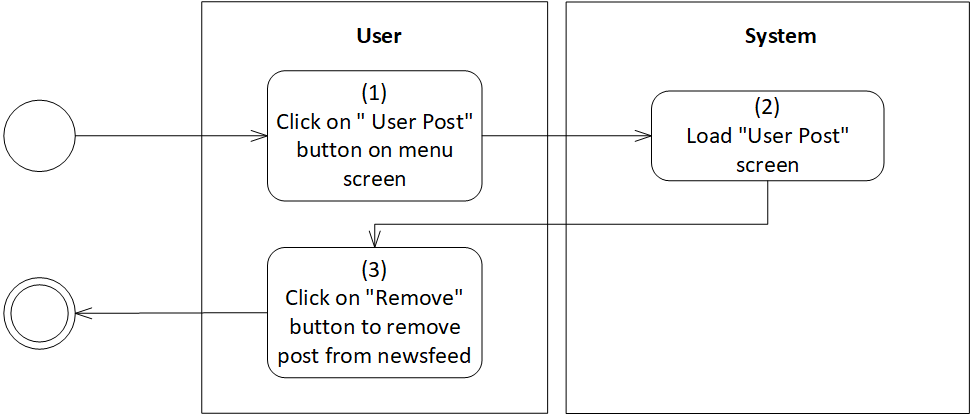
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Post” screen rules**:  When user clicks on “User Post” button, the system loads “User Post” screen. (Refer to “User Post” screen in “UI Description” file) |
| *(6)* | *BR2* | **View post rules:**  User posts will be displayed as the template below:   | Username | Get [Username] of user | | --- | --- | | Avatar | Get [Avatar] of user | | Body | [Body] = [Status] of user | |

# Approve user post

| **Name** | **Approve user post** |
| --- | --- |
| **Description** | This use case allows administrator to approve user posts. |
| **Actor** | Administrator |
| **Trigger** | * Click on “User Post” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “User Post” button |
| **Post-condition** | * Remove from the general wall. |

**Activity Flow**



*Figure 72: Activity Flow - Approve post*

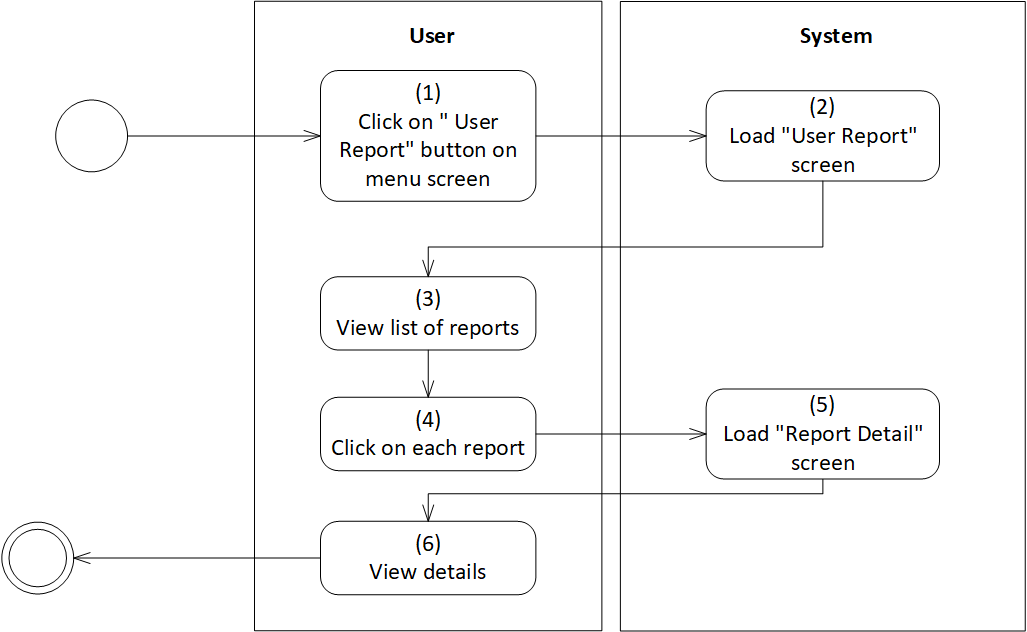
**Business Rules:**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Post” screen rules**:  When user clicks on “User Post” button, the system loads “User Post” screen. (Refer to “User Post” screen in “UI Description” file) |
| *(3)* | *BR2* | **Approve post rules**:  If the posts of user violate the application rules, administrator can click on “Remove” button to remove these posts. After that, the system will load MSG3. |

# View user reports

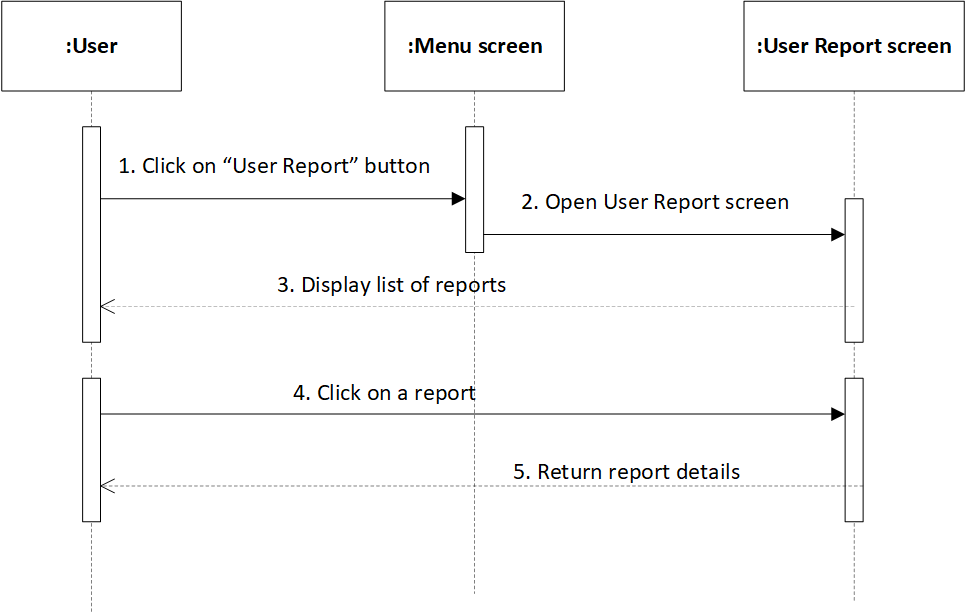
| **Name** | **View user reports** |
| --- | --- |
| **Description** | This use case allows administrator to view reports from all users in the system. |
| **Actor** | Administrator |
| **Trigger** | * Click on “User Report” button on menu |
| **Pre-condition** | * Must be connected to the network * Open the application * Have signed up already * Access menu screen * Click on “User Report” button |
| **Post-condition** | * View each report from user. |

**Activity Flow**



*Figure 73: Activity Flow - View user reports*

**Sequence Diagram**



*Figure 74: Sequence Diagram - View user reports*

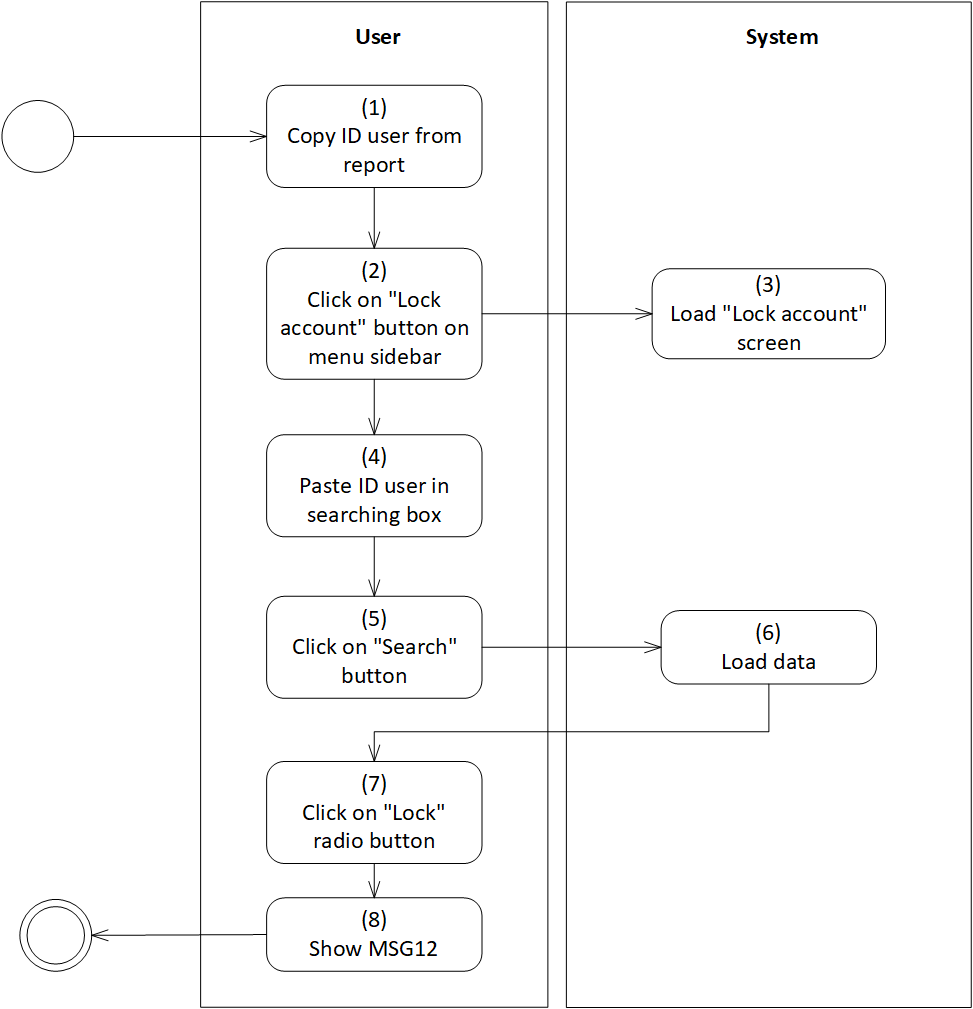
**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Report” screen rules**:  When user clicks on “User Report” button, then clicks on “User Report” button, the system loads “User Report” screen that contains list of reports (Refer to “User Report” screen in “UI Description” file) |
| *(5)* | *BR2* | **Load “Report Details” Form rules**:  When user clicks on each report, the system loads “Report Details” form (Refer to “Report Details” list in “UI Description” file) |
| *(6)* | *BR3* | **View report details rules:**  User report details will be displayed as the template below:   | Reporter | Get [Username] of reporter | | --- | --- | | Reported User | Get [Username] of user reported | | ID | Get [ID] of user reported | | Body | [Body] = [Reason] + [Other] of reporter | |

# Lock account

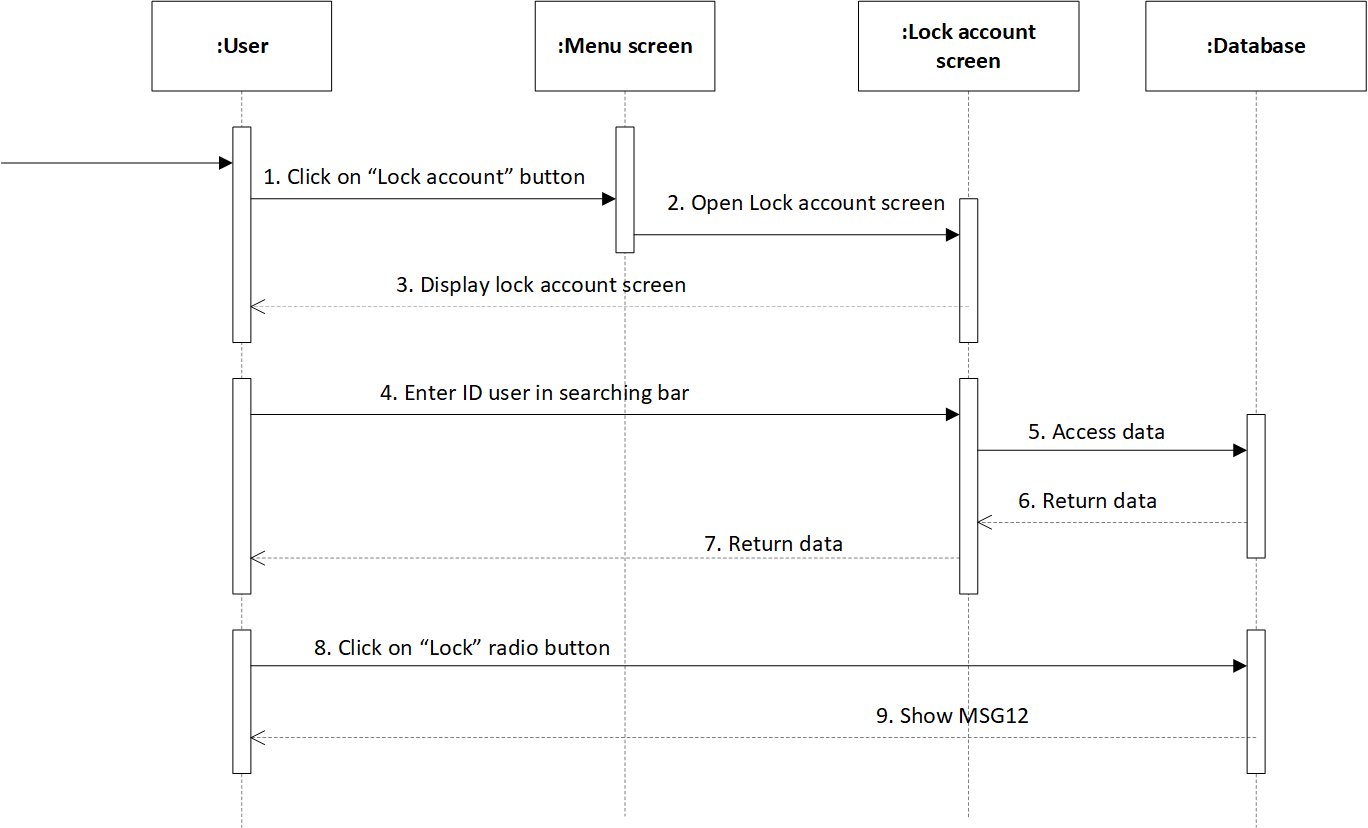
| **Name** | **Lock account** |
| --- | --- |
| **Description** | This use case allows administrator to lock user accounts. |
| **Actor** | Administrator |
| **Trigger** | * Open the application * Click on “Lock account” button on menu |
| **Pre-condition** | * Must be connected to the network * Have signed up already * Access menu screen * Click on “Lock account” button on menu |
| **Post-condition** | * The account is locked |

**Activity Flow**



*Figure 75: Activity Flow - Lock account*

**Sequence Diagram**



*Figure 76: Sequence Diagram - Lock account*

**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(3)* | *BR1* | **Load “Lock Account” screen rules**:  When user clicks on “Lock Account” button, then clicks on “Lock Account” button, the system loads “Lock Account” screen that contains list of reports (Refer to “Lock Account” screen in “UI Description” file) |
| *(6)* | *BR2* | **Load data rules**:  When user enters user ID in searching box and clicks on “Search” button, the system will retrieve [ID] in the database so that [ID] in database = [ID] in searching box.   * If found, the system load data related to the user ID. * Else, set [Text Error] = “No user is found” |
| *(7)* | *BR3* | **Lock account rules:**  When user clicks on “Lock” radio button, the system will load MSG12 |

# Send warning to user

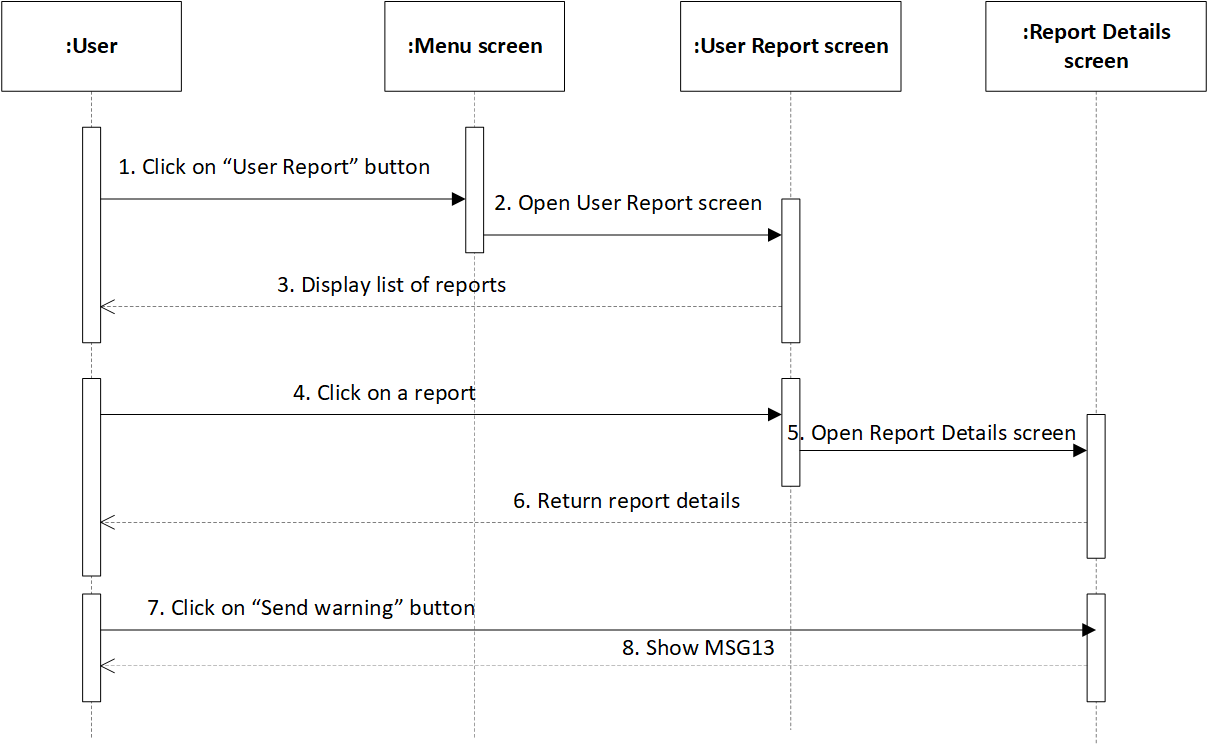
| **Name** | **Send warning to user** |
| --- | --- |
| **Description** | This use case allows administrator to send warning to user who is reported by other users. |
| **Actor** | Administrator |
| **Trigger** | * Open the application * Click on “User Report” button on menu |
| **Pre-condition** | * Must be connected to the network * Have signed up already * Access menu screen * Click on “User Report” button on menu |
| **Post-condition** | * Send warning successfully |

**Activity Flow**



*Figure 77: Activity Flow - Send warning to user*

**Sequence Diagram**



*Figure 78: Sequence Diagram - Send warning to user*

**Business Rules**

| **Activity** | **BR Code** | **Description** |
| --- | --- | --- |
| *(2)* | *BR1* | **Load “User Report” screen rules**:  When user clicks on “User Report” button, the system loads “User Report” screen that contains list of reports (Refer to “User Report” screen in “UI Description” file) |
| *(5)* | *BR2* | **Load “Report Details” screen rules**:  When user clicks on each report, the system loads “Report Details” screen (Refer to “Report Details” list in “List Description” file) |
| *(6)* | *BR3* | **View report details rules:**  User report details will be displayed as the template below:   | Username | Get [Username] of reporter | | --- | --- | | Username | Get [Username] of user reported | | ID | Get [ID] of user reported | | Body | [Body] = [Reason] + [Other] of reporter | |
| *(6)* | *BR4* | **Send warning rules:**  When user clicks on “Send warning” button,   * The system will load MSG13 * The system send warning as the template below:  | Username | Get [Username] of user reported | | --- | --- | | ID | Get [ID] of user reported | | Body | [Body] = “WARNING”  [Body] = [Body] + new 2 lines  [Body] = [Body] + “Someone has just reported you with administrator group”  [Body] = [Body] + new line  [Body] = [Body] + “We hope” + [Username] + will follow our community rules. If this reporting process continues happening many times, we have to lock your account.” | |

# UI Description



Link Figma: <https://www.figma.com/file/4ASNECJpRCWNI8WmWTsdzS/Dating-App?node-id=5%3A1968>

# Database Description



# Non-functional Requirements

# User Access and Security

| **Group**  **Function / Data** | **Normal User** | **Silver User** | **Gold User** | **Administrator** | **System Timer** |
| --- | --- | --- | --- | --- | --- |
| **Manage “Account”** |  |  |  |  |  |
| Sign in | X | X | X | X |  |
| Sign up | X | X | X |  |  |
| Sign out | X | X | X | X |  |
| Remember account | X | X | X | X |  |
| Forgot password | X | X | X |  |  |
| Change password | X | X | X |  |  |
| Upgrade account | X | X | X |  |  |
| Block account |  |  | X |  |  |
| Lock account |  |  |  | X |  |
| Unblock account |  |  | X |  |  |
| Boost account |  |  | X |  |  |
| **Manage “Digital Wallet”** |  |  |  |  |  |
| Link bank account |  |  |  | X |  |
| Deposit money into digital wallet | X | X | X | X |  |
| **Manage “Notification”** |  |  |  |  |  |
| Get notification | X | X | X |  |  |
| Send warning |  |  |  | X |  |
| **Manage “Matching”** |  |  |  |  |  |
| Send matching request | X | X | X |  |  |
| Receive matching request | X | X | X |  |  |
| View matching history |  | X | X |  |  |
| Match with searching user |  | X | X |  |  |
| **Manage “Entertainment room”** |  |  |  |  |  |
| Create chat room | X(\*) | X(\*) | X(\*) |  |  |
| View chatting history | X(\*) | X(\*) | X(\*) |  |  |
| Chat | X(\*) | X(\*) | X(\*) |  |  |
| Create video call room |  | X(\*) | X(\*) |  |  |
| View calling history |  | X(\*) | X(\*) |  |  |
| **Manage “User Information”** |  |  |  |  |  |
| View general user information | X | X | X |  |  |
| View user information |  |  |  | X |  |
| Edit personal information | X | X | X |  |  |
| **Manage “Post”** |  |  |  |  |  |
| Create, Remove |  | X(\*) | X(\*) |  |  |
| Edit |  | X(\*) | X(\*) | X |  |
| View | X | X | X | X |  |
| **Manage “Report”** |  |  |  |  |  |
| Report user |  | X | X |  |  |
| Report incident |  | X | X | X |  |
| View user report |  |  |  | X |  |
| Filter users | X | X | X |  |  |
| Search user |  | X | X |  |  |
| Add to the Favorites list |  |  | X |  |  |
| View Favorites list |  |  | X |  |  |
| Approve user post |  |  |  | X |  |
| Review app | X | X | X |  |  |

X: User has full permission to do the action.

X(\*): User only has permission to do the action on his own items

# Performance Requirements

**Number of users:**

* Number of concurrent users: 500
* Number of users the application can handle: 2000

**Data volume**

* Number of documents: 5MB ~ 15 file
* Data growth rate: 5MB/ day

**Level of availability**

Availability level required for this application: 24\*7

**Usage frequency**

Usage frequency: Daily

# Implementation Requirements

**Location: AWS /AZURE/SERVER**

The location where this website will be deployed: Vietnam

**Read-only Duration**

From 4 - 6 hours

**Read-only Timeframe**

00:00 AM

**Maintenance Window**

This application will be maintained: Monthly

**UI Wireframe**

Link Figma:<https://www.figma.com/file/4ASNECJpRCWNI8WmWTsdzS/Dating-App?node-id=5%3A1968>

**Other plans and activities**

N/A

# Other Requirements

# Archive Function

[Enable Archival Function for following list:

| **List** | **Actor** | **Condition** |
| --- | --- | --- |
| *List name* | *Actor name* | *Actor* is able to archive item in “*list name*” list by created date. |

[For details, refer to section 6.4 Reference.]

# Security Audit Function

[Enable Security Audit Function for {Actor name} to tracking any modification on user’s permission.]

# Các yêu cầu hệ thống

## Sites

| **#** | **Site Name** | **Description** |
| --- | --- | --- |
| *1* | ***TLCo.com*** |  |
| *2* | ***Email.Tlco.com*** |  |
| *3* | ***Docs.tlco.com*** |  |

## Lists

| **#** | **List Code** | **List Name** | **Description** |
| --- | --- | --- | --- |
| *1* |  |  |  |
| *2* |  |  |  |
| *3* |  |  |  |

## Permission Levels

| **#** | **Permission Level** | **Permissions** | **Description** |
| --- | --- | --- | --- |
| *1* | *Full Control* |  |  |
| *2* | *Contribute* |  |  |
| *3* | *Update XXX* |  |  |

## Groups

| **#** | **Group Name** | **Permission Level** | **Description** |
| --- | --- | --- | --- |
| *1* | *Admin* |  |  |
| *2* | *Requestor* |  |  |
| *3* | *….* |  |  |

## Web Parts

| **#** | **Web Part** | **Description** |
| --- | --- | --- |
| *1* | ContainerWebPart | * To host custom user controls (belong to Framework source code) |
| *2* | ArchiveEditWebPart | * To implement the archive function (belong to Framework source code) |
| *3* | ArchiveListByAgentWebPart | * To implement the archive function (belong to Framework source code) |
| *4* | ArchiveListWebPart | * To implement the archive function (belong to Framework source code) |
| *5* | AuditSettingsWebPart | * To implement the audit function (belong to Framework source code) |
| *6* | AuditViewerWebPart | * To implement the audit function (belong to Framework source code) |
| *7* | DialogListWebPart | * To implement the dialog control (belong to Framework source code) |
| *8* | PrintPreviewWebPart | * To implement the print function (belong to Framework source code) |
| *9* | QuickLaunchContextWebPart | * To implement the quick launch control (belong to Framework source code) |
| *1* | QuickLaunchItemWebPart | * To implement the quick launch control (belong to Framework source code) |
| *1* | QuickLaunchManagerWebPart | * To implement the quick launch control (belong to Framework source code) |

## Custom Pages

[If there is no custom page, remove the table and use this sentence:

There is no custom page implemented in this application.]

| **#** | **Page Name** | **Description** |
| --- | --- | --- |
| *1* |  |  |
| *2* |  |  |
| *3* |  |  |

## Scheduled Agents

[If there is no scheduled agent, remove the table and use this sentence:

There is no scheduled agent implemented in this application.]

| **No.** | **Name** | **Description** | **Rule** | **Agent Main Class** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Technical Concern

[List all factors that can affect the performance of application such as

* Huge amount of data => saving/loading issue
* Too much content is in a single page
* Link with banks to deposit money

# Appendixes

# Glossary

The list below contains all the necessary terms to interpret the document, including acronyms and abbreviations.

| **Term** | **Description** |
| --- | --- |
| *BR* | **B**usiness **R**ule |
| *CBR* | **C**ommon **B**usiness **R**ule |
| *DB* | Notes **D**ata**b**ase |
| *MSG* | **M**es**s**a**g**e |
| *UC* | **U**se **C**ase |
| *N/A* | **N**ot **A**vailable or **N**ot **A**pplicable, used to indicate when information in a certain section could not be provided because it does not apply to this application. |
| *UI* | **U**ser **I**nterface |
| *SRS* | **S**oftware **R**equirements **S**pecification |
| *TBD* | **T**o **b**e **d**etermined or **t**o **b**e **d**efined |

# Mapping to Notes Application

This section describes the mapping between the migrated application and its source Notes application, including the mapping for data objects, features, actors.

| **Migrated Application’s Elements** | **Notes Application’s Elements** |
| --- | --- |
| ***Actors*** | |
|  |  |
|  |  |
|  |  |
| ***Use Cases / Features*** | |
|  |  |
|  |  |
|  |  |
|  |  |
| ***Lists*** | |
|  |  |
|  |  |
|  |  |
| ***Views*** | |
|  |  |
|  |  |
|  |  |

# Messages

This section describes the details of messages used in business rules

E.g: error messages, confirmation messages, etc.,

| **Message Code** | **Message Content** | **Button** |
| --- | --- | --- |
| MSG0 | “The application failed to conduct this operation.” | OK |
| MSG1 | “Are you sure you want to exit the application?” | YES/NO |
| MSG3 | “The application succeeded in conducting this operation.” | OK |
| MSG4 | “You have deposited money successfully!” | OK |
| MSG5 | “You have sent a matching request successfully!” | OK |
| MSG6 | “You have created a chat room successfully!” | OK |
| MSG7 | "Thank you for rating our application. We will update it to bring a better experience to users soon.” | OK |
| MSG8 | “You have reported the user successfully!” | OK |
| MSG9 | "You have created a video call room successfully!” | OK |
| MSG10 | “You have just blocked a user. We will not notify them about your decision.” | OK |
| MSG12 | “You have locked account successfully!” | OK |
| MSG13 | “You have sent a warning successfully!” | OK |

# Issues List

N/A