NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

SOFTWARE REQUIREMENTS SPESIFICATIONS

Implement extensions
on
JAIN-SIP-PRESENCE-PROXY
and
SIP Communicator

Release:	1.0
Date of Print:	
Date of Release:	November, 2016
State of Release:	Initial
State of Approval:	Draft
Approved by:	Kontogiannis Konstantinos
Prepared by:	Kallas Konstantinos, Karampasi Aikaterini, Tzinis Eythymios,
Inspected by:	Papaspyrou Nikolaos
Archive's Name:	
Form's Number:	

Table of Contents

Introduction

Purpose

This document presents in detail the software requirements specifications of the implementation of extensions on JAIN-SIP-PRESENCE-PROXY and SIP Communicator applications.

Overview

JAIN-SIP-PRESENCE-PROXY application is an implementation of SIP Proxy Server, SIP Registrar Server, which uses the libraries of JAIN SIP libraries software and acts simultaneously as a SIP Location Server. SIP Communicator is an IP phone implemented on software. On this project, we're going to implement some extensions in order to allow the following:

a. For the user to be able to create a list of numbers he does not want to receive calls from (call blocking). When he receives a call from someone who's in the list the call will be automatically rejected.
b. Charging according to the pricing policy. Only he who calls will be charged.
c. For a user to be able to forward a call to another user (call forwarding).

Those extensions must be compatible to the prototype RFC 3261.

Business Scenario Model

Actors

Overview

The SIP protocol architecture depends on four modules: the SIP Registrar server, SIP Proxy servers, SIP Location Servers and SIP User Agents. Specifically, the SIP Registrar servers allow to the users to make their registration and let the system know their location so that other users know they're there. Users may contact the SIP Registrar servers through the SIP Communicator which is basically another one SIP User Agent. The SIP Proxy servers allow the routing of the requests being made, in order to verify the presence of a user in the system, the implementation of the routing and connecting the calls between the users and also to verify the users' credentials so that certain services will be executed. SIP Location servers check the on-line status and the user's specific location of the connection (address of the exact Registrar server).
Last but not least, SIP User Agents allow the users' access into the system (registration, delete, make a call to other users etc.). SIP User Agents are the system's endpoints of a call.

Actor Diagram

Actor Definition

Non-Authenticated user:

Description	Someone who wants to use this way of communicating with others will have to be registered to the system. Having said that, we mean that a user must request to be registered and then authenticated with his credentials. As long as the user is not authenticated he cannot make any calls.
Aliases	None
Inherits	None
Actor Type	Passive Person
Contact Person	Papaspyrou Nikolaos
Contact Details	kkontog@softlab.ntua.gr

Authenticated user:

Description	A user has requested to be registered and he has been authenticated by the system with his credentials.
Aliases	None
Inherits	None
Actor Type	Active Person
Contact Person	Papaspyrou Nikolaos
Contact Details	kkontog@softlab.ntua.gr

System:

Description	The system appears to be the entity which consists of SIP Communicator and JAIN-SIP-PRESENCE-PROXY. These will decide if it will let the user enter the communicator and make calls to other users.
Aliases	SIP Communicator and JAIN-SIP-PRESENCE-PROXY
Inherits	None
Actor Type	External System
Contact Person	Papaspyrou Nikolaos
Contact Details	kkontog@softlab.ntua.gr

Use Case Descriptions

In the following section we present the Use Cases as mentioned in section 2 of the project description. These are discriminated in three phases. First of all, we describe the registering scenario, which happens only the first time the user enters the system. After that, we analyze some cases in which there is no problem and some cases in which there some kind of "catch" to be careful of, which we'll call pathological.

The scenarios mentioned are code-named as mentioned below:

Use Case-<stateOfScenario>-<UseCaseNumber>-<NameOfUseCase>

where:

<stateOfScenario>: if it's referring to a normal or a pathological scenario or if it's the registering scenario

<UseCaseNumber>: the number of the case as mentioned in the project description

<NameOfUseCase>: the description given as the name from the project description

Below we describe in more detail those scenarios.

Use Case Register 1: First Registration on the system

Before we can use any of the things that this application provides us we have to be registered and the system to authenticate us.

Description

The first thing that needs to be done is to write down our personal information that are needed for the system to provide to us it's services.

Actors

Non authenticated users.

Requirements & Caveats

1. Download the application
2. Internet connection

Use

Use-Case-Register-1.1: User A registers on the Proxy Server by using a username and a password.

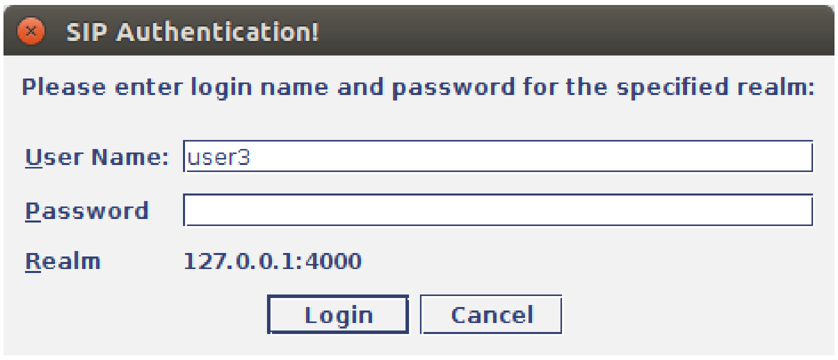
Alternative Use Case

N/A

Extends

N/A

User Interface


SIP Authentication!

Please enter login name and password for the specified realm:

User Name: user3

Password

Realm 127.0.0.1:4000

Questions

N/A

Notes

N/A

Authors

Konstantinos Kallas

References

N/A

Use Case Normal 1: Phone Call without Forwarding and Restrictions (Normal Call)

This is the most fundamental use case of the system. This scenario applies to all Authenticated Users.

Description

The purpose of this Use Case is the audio-visual communication between two Users who are both registered on the Proxy Server.
The result of this use case is the successful communication between the two users.

Actors

Authenticated User, System

Requirements & Caveats

1. Both users are required to be registered in the system in order the scenario to be plausible.

Use Case Description

Considering that User A wants to call User B.

Use-Case-1.1: User A writes User's B username and clicks Dial in order to initialize the call.

Use-Case-1.2: User B clicks Answer and the call starts.

Use-Case-1.3: Users click Hangup and the call ends.

Alternative Use Case

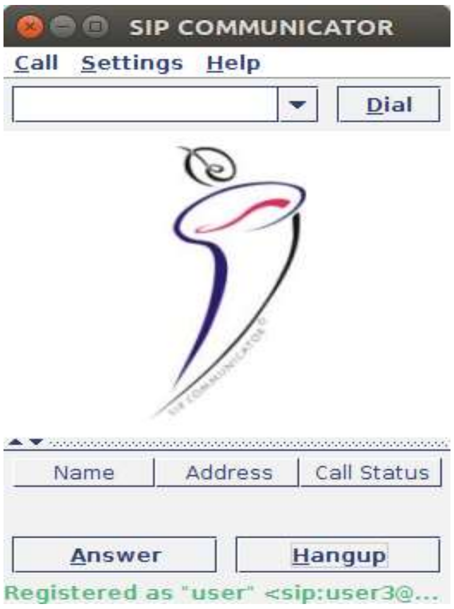
Use-Case-1.2.B: User B clicks Hangup and the call doesn't start. User A receives a "Busy" message.

Extends

N/A

User Interface

The main Interface. (The "Dial", "Answer" and "Hangup" buttons are visible)



Registered as "user" <sip:user3@...

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Normal 2: Block certain incoming calls

Description
In this scenario it is studied an example in which a user of our system desires to prevent calling from specific users. Each user has the ability to fill in a list with designated users of whom he will not be able to accept calls. For example, the user X wants to prevent the user Y from calling him. The user X adds the user Y in the aforementioned list and the later will always see the status of the former as “Unavailable” without knowing his true status.

Actors
1. A user Y who has to be prohibited from calling another user X.
2. The user X who wants to block user Y from calling him.
3. System which enables the user X to edit his list of blocked users and prohibit the user Y from calling him.

Requirements & Caveats
1. Both users are required to be registered in the system in order the scenario to be plausible.
2. The user who desires to block incoming calls has to sign in the system with valid credentials.

Use Case Description
We speculate that the user X wants to block the user Y from calling him and both of them are registered in the system.

Use-Case-2.1: User X authenticates with his credentials in the system.

Use-Case-2.2: The user X selects the blocked list from the respective button displayed in the GUI.

Use-Case-2.3: The user X can edit his list of blocked usernames by adding new names in it. The names written in the list are obviously prohibited from calling the user.

Use-Case-2.4: The user X has to click apply to save the changes.

Alternative Use Case
The user X can also delete some prior existing usernames in his blocked list. Hecan easily do that by the selection of a username, click the “Delete” button and the click “Apply”.

Extends
N/A

User Interface
N/A

Questions
N/A

Notes
The implementation will follow the specifications above but it is not ready yet so we cannot provide any relevant information about the User Interface.

Authors
Eythymios Tzinis

References
N/A

Use Case Normal 3: Charging the call
After each call the caller is charged with a already specified amount based on the call duration.

This use case happens after every call when either the caller or the callee hangs up.

Description
The purpose of this use case is the billing of the users when they use the application.

The result is a money transfer from the user’s bank account to the company’s bank account. The amount of money transferred changes based on the call duration.

Actors
Authenticated User, System

Requirements & Caveats
1. Both users are required to be registered in the system in order the scenario to be plausible.
2. The users have to provide their billing information (Bank Account, Billing Address)

Use Case Description
Considering that User A is the caller and User B is the callee

Use-Case-3.1: User A or User B hangs up the call

Use-Case-3.2: The Proxy Server calculates the amount to be transferred

Use-Case-3.3: (Out of the system) User A is charged the specified amount

Use-Case-3.4: (Out of the system) The amount is deposited at the company's bank account

Alternative Use Case
Use-Case-1.3.B: User B clicks Hangup and the call doesn't start. User A receives a "Busy" message.

Extends
N/A

User Interface
This Use Case is not bound to any specific User Interface.

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Normal 4: Call Forwarding

Description
In this scenario a call forwarding is demonstrated. In particular, the user X wants to forward all his incoming calls into another username e.g. user Y. Consequently a call from another user A to user X has to be redirected by the system to the user Y without ringing the phone of user X. The user Y can now establish a connection with the user A. Thus, multiple forwards are allowed by our system. The chained reaction of redirection of the first call is implemented in our system which can discern the differences between a valid and not circular forwarding schema and a circular. Always the last person in the forwarding chain is the final receiver of the call. Some schemes are presented:

Simple Forwarding (Mentioned above - Valid): user A→user X→ user Y

Multiple Forwarding (No Repetition of Users - Valid): user A→user X→ user Y → … → user Z

Circular Forwarding (Invalid): user A→user X→ user Y → … → user A

Circular Forwarding (General Case - Invalid): user X1→ user X2 → … →user XN→ user Xj

where j = 1,2,…,N

Actors

Without perturbing the general form of forwarding schema we can describe only the simple case. The other cases are derived easily by this one by inducing the following thinking.

1. A user A who makes the initial call to user X.
2. The user X who has selected to forward his incoming calls to the user Y.
3. User Y who is the final receiver of the first call.
4. System which is essential for offering all this functionality.

Requirements & Caveats

1. Both users are required to be registered in the system in order the scenario to be plausible.
2. The users who desire to make a forwarding chain have to take care for not generating an invalid circular schema as mentioned before. In this case the call should be prohibited and show the first user of the call an apt error message by simultaneously closing the connection.

Use Case Description

There are 3 different acts in this scenario:

Caller & Receiver:

Same as mentioned in the 1st use case.

User who wants to forward his calls:

Use-Case-4.1: User X authenticates with his credentials in the system.

Use-Case-4.2: The user X selects from the GUI the button “Options”.

Use-Case-4.3: The user X selects from the list “Forwarding”.

Use-Case-4.4: The user can now type the user Y who wants to forward his incoming calls to.

Use-Case-4.5: If the user typed is valid then the system offers this attribute freely without warning.

Use-Case-4.6: The user X has to click apply to save the changes.

Alternative Use Case

N/A

Extends
N/A

User Interface
N/A

Questions
N/A

Notes
The implementation will follow the specifications above but it is not ready yet, so we cannot provide any relevant information about the User Interface.

Authors
Eythymios Tzinis

Use Case Pathological 1: Not Connected User B
Sometimes a User's information cannot be located by the server. When this happens the system should act according to the RFC 3261 standard.

Description
The purpose of this use case is to handle a problematic situation in a healthy way according to a standard.

The result is described in the RFC 3261 standard.

Actors
Authenticated User, System

Requirements & Caveats
1. User A has to be registered and signed in.
2. User B should't be located by the server.

Use Case Description
Considering that User A is the caller and User B is the callee

P-Use-Case-3.1: User A attempts to call User B

P-Use-Case-3.2: The server is unable to locate User B information

P-Use-Case-3.3: The case is handled according to RFC 3261

Alternative Use Case
N/A

Extends
N/A

User Interface
The User Interface is the same as the one seen in Use-Case-Normal-1 with the addition of a message describing that User B could not be located.

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Pathological 2: User A SIP Communicator Crash
Sometimes User's A SIP Communicator could crash. In this case the system should be able to handle the situation according to RFC 3261 standard as above.

Description
The purpose of this use case is to handle a problematic situation in a healthy way according to a standard.

The result is described in the RFC 3261 standard.

Actors
Authenticated User, System

Requirements & Caveats
1. User A and User B have to be registered and signed in.
2. User A has already called User B.

Use Case Description
Considering that User A is the caller and User B is the callee

P-Use-Case-2.1:User A calls User B as is described in Use-Case-1

P-Use-Case-2.2: User A SIP Communicator crashes unexpectedly

P-Use-Case-2.3: The case is handled according to RFC 3261

Alternative Use Case
N/A

Extends
N/A

User Interface
The User Interface for User B is the same as the one seen in Use-Case-1 with the addition of a message describing that User A SIP Communicator has crashed.

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Pathological 3: User B SIP Communicator Crash
Sometimes User's B SIP Communicator could crash. In this case the system should be able to handle the situation according to RFC 3261 standard as above.

Description
The purpose of this use case is to handle a problematic situation in a healthy way according to a standard.

The result is described in the RFC 3261 standard.

Actors
Authenticated User, System

Requirements & Caveats
1. User A and User B have to be registered and signed in.
2. User A has already called User B.

Use Case Description
Considering that User A is the caller and User B is the callee

P-Use-Case-2.1:User A calls User B as is described in Use-Case-1

P-Use-Case-2.2: User B SIP Communicator crashes unexpectedly

P-Use-Case-2.3: The case is handled according to RFC 3261

Alternative Use Case
N/A

Extends
N/A

User Interface
The User Interface for User A is the same as the one seen in Use-Case-Normal-1 with the addition of a message describing that User B SIP Communicator has crashed.

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Pathological 4: Proxy Server Crash
Sometimes a User might be registered normally and then the Proxy Server unexpectedly crashes. The system should be able to handle the situation according to RFC 3261 as above and also notify the User of the Proxy Crash.

Description
The purpose of this use case is to handle a problematic situation in a healthy way according to a standard.

The result is described in the RFC 3261 standard.

Actors
Authenticated User, System

Requirements & Caveats
1. User A has to be registered and signed in.

Use Case Description
P-Use-Case-4.1: Proxy server crashes

P-Use-Case-4.2: The case is handled according to RFC 3261

Alternative Use Case
N/A

Extends
N/A

User Interface
The User Interface for User A is the same as the one seen in Use-Case-1 with the addition of a message describing that Proxy Server has crashed.

Questions
N/A

Notes
N/A

Authors
Konstantinos Kallas

References
N/A

Use Case Diagrams
On the following images we present the use cases diagrams as mentioned above:

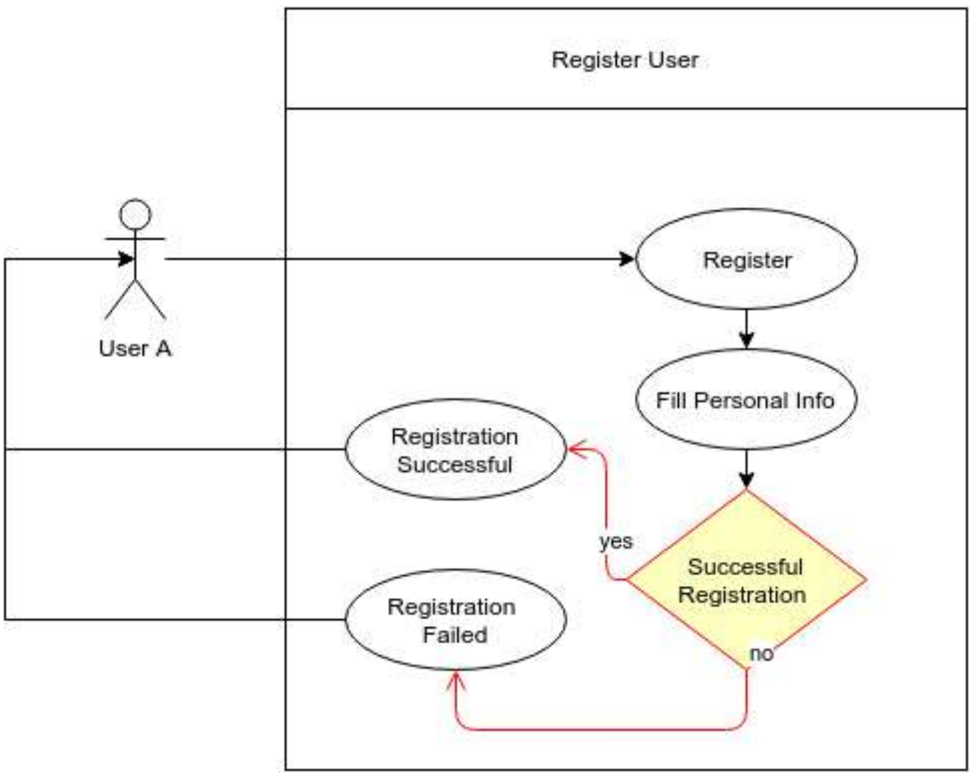


Image 1: Registering

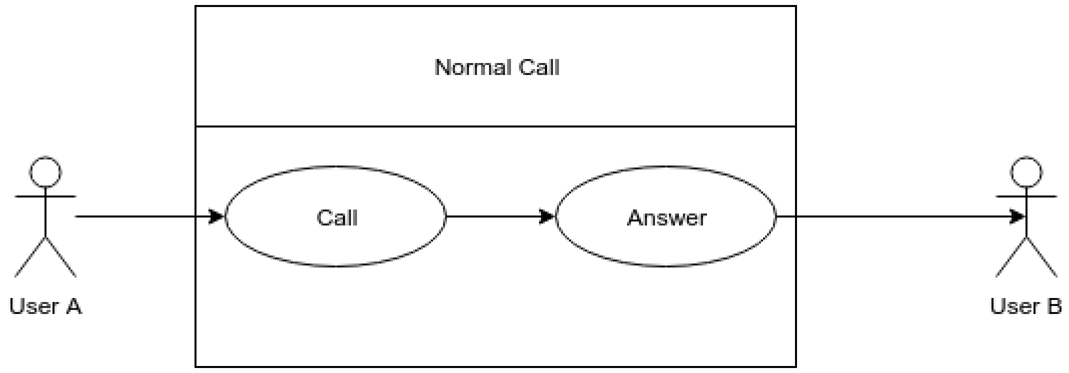


Image 2: Normal Call

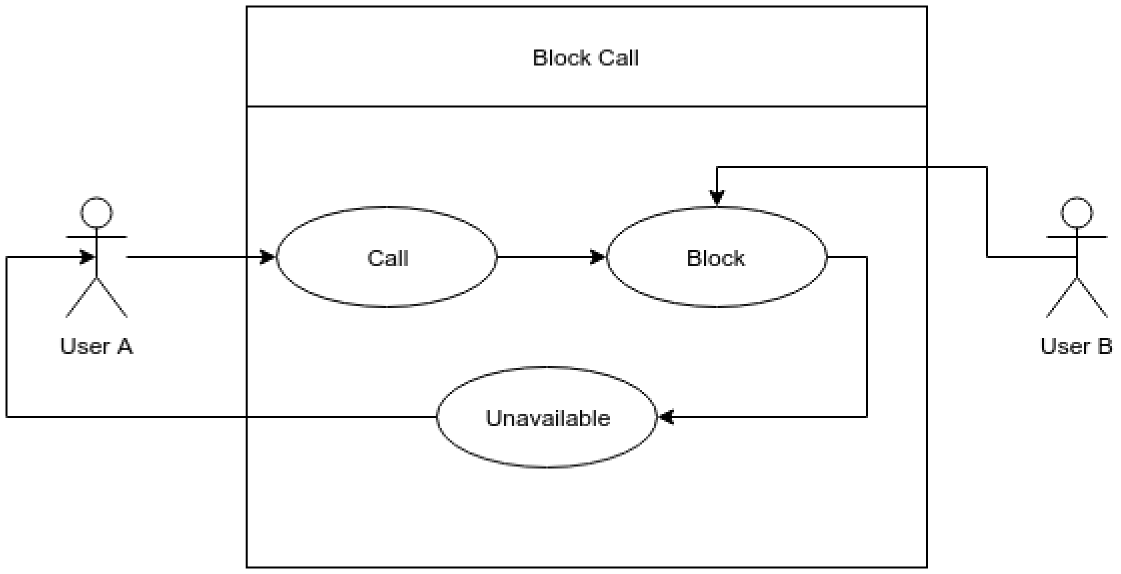


Image 3: Block Incoming Call

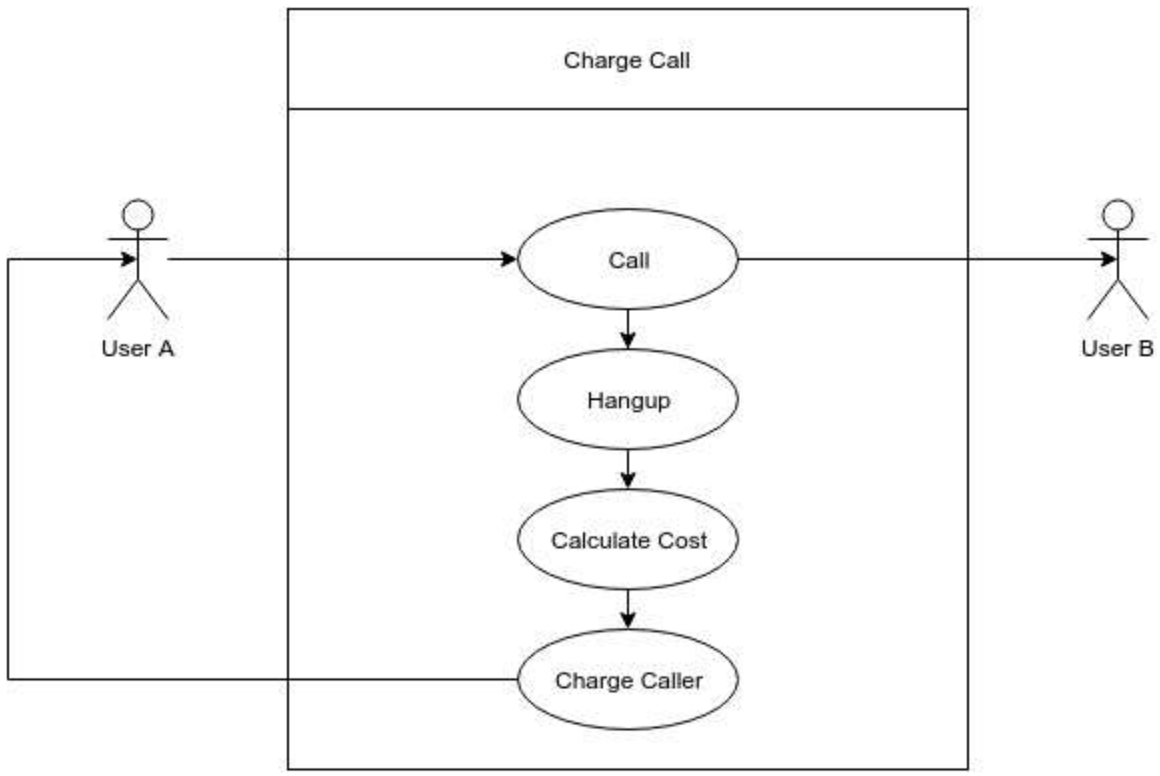


Image 4: Charge Call

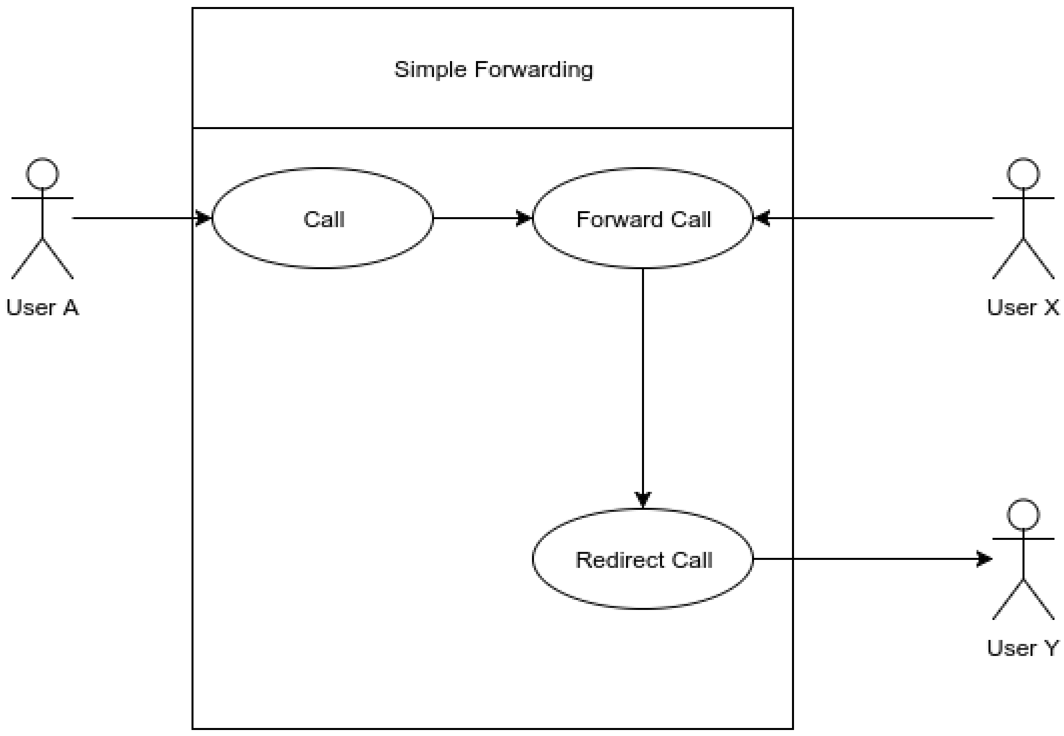


Image 5: Forwarding Call

As far as it concerns forwarding we have to provide two diagrams concerning three different ways of forwarding a call. General circular forwarding contains the two scenarios of circular forwarding mentioned in the project description.

1.

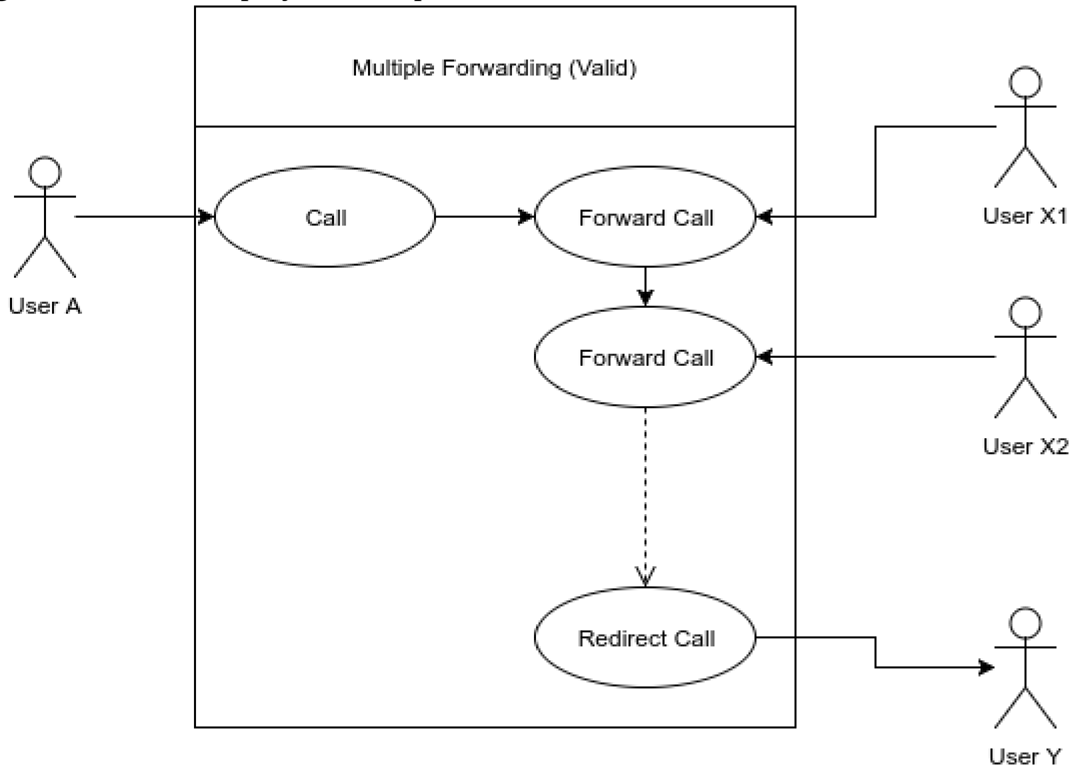


Image 6: Multiple Forwarding

2.

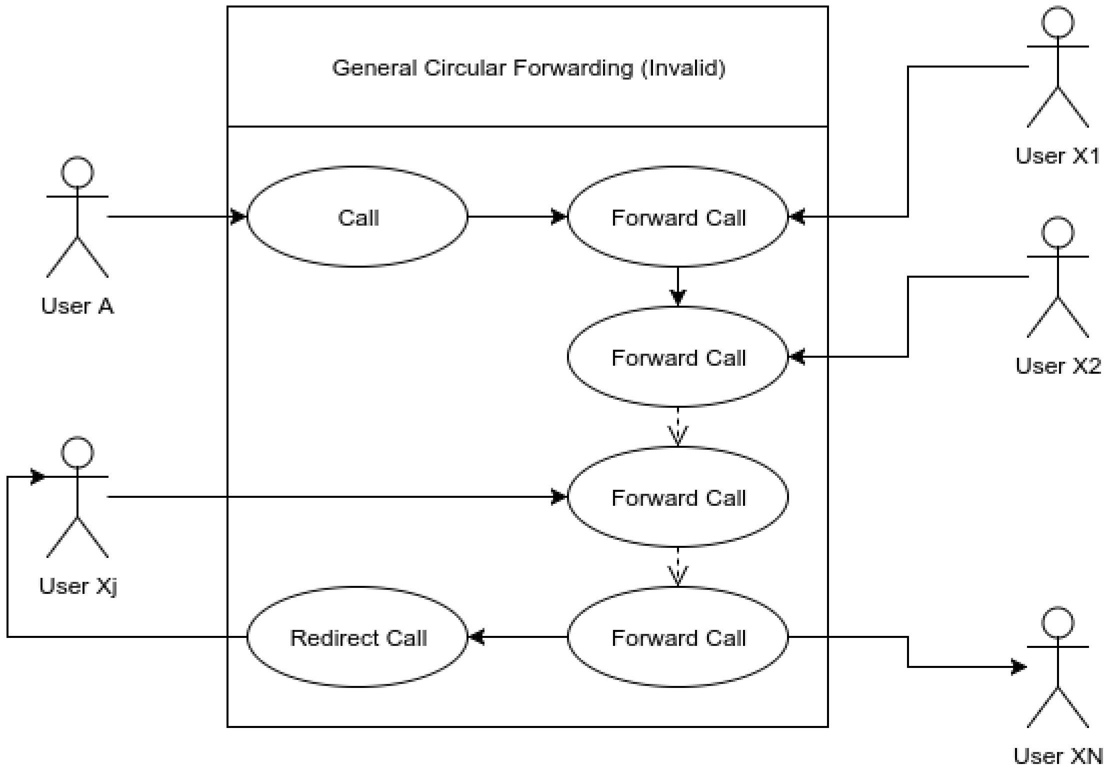


Image 7: General Circular Invalid Forwarding

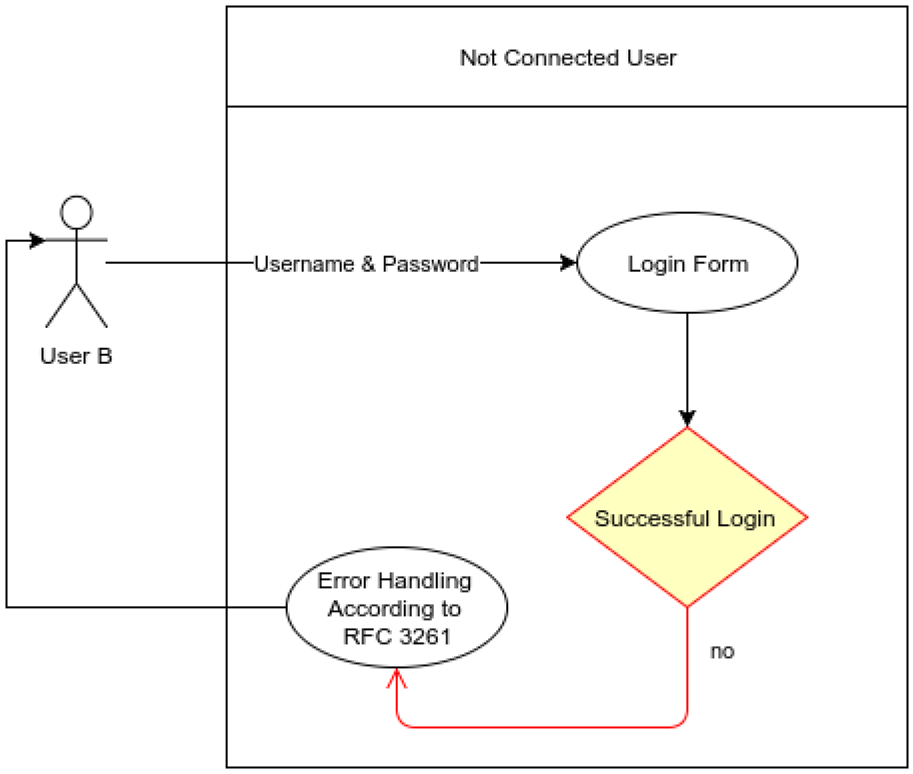


Image 6: Not Connected User B

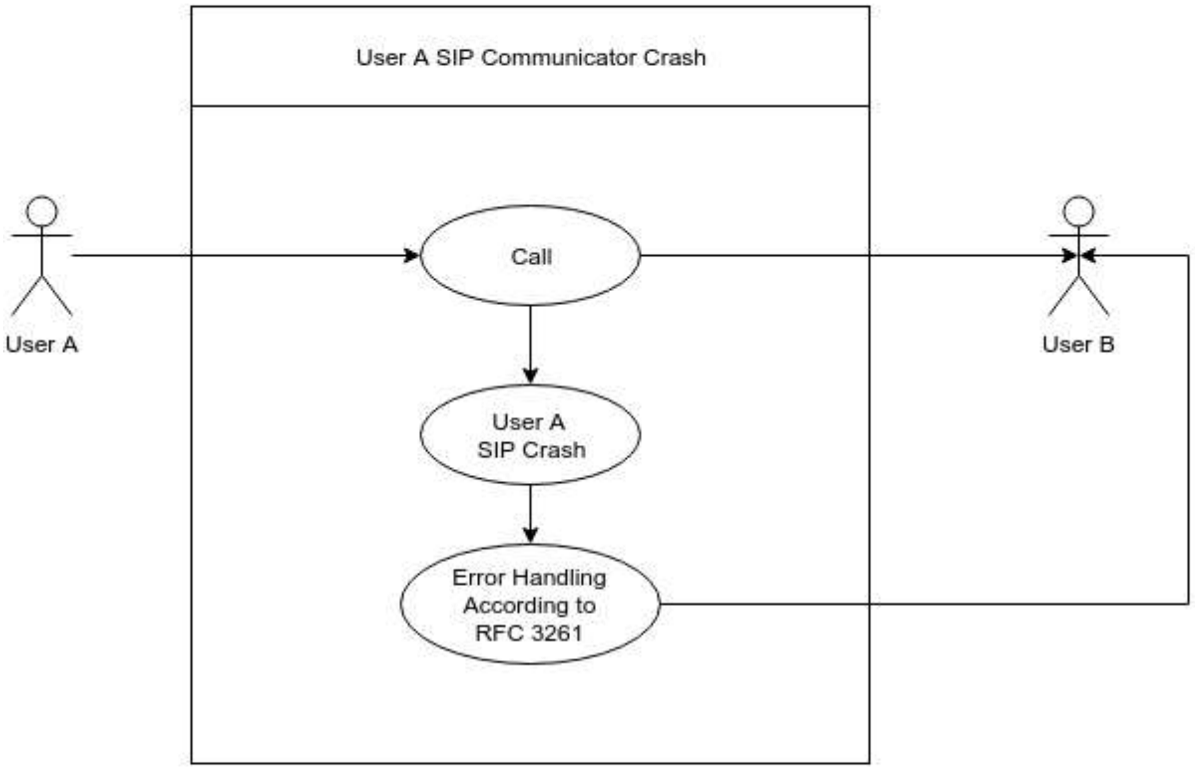

User A SIP Communicator Crash
User A
Call
User B
User A SIP Crash
Error Handling According to RFC 3261

Image 7: User A SIP Crashed

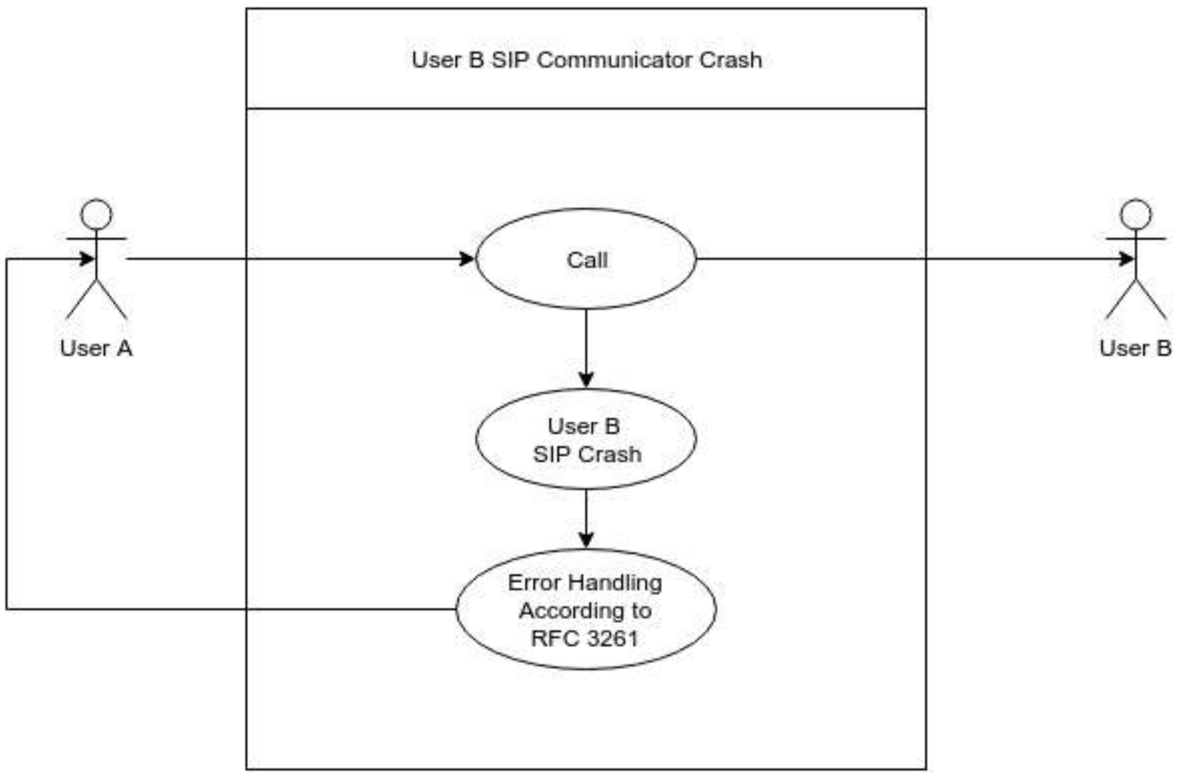


Image 8: User B SIP Crashed

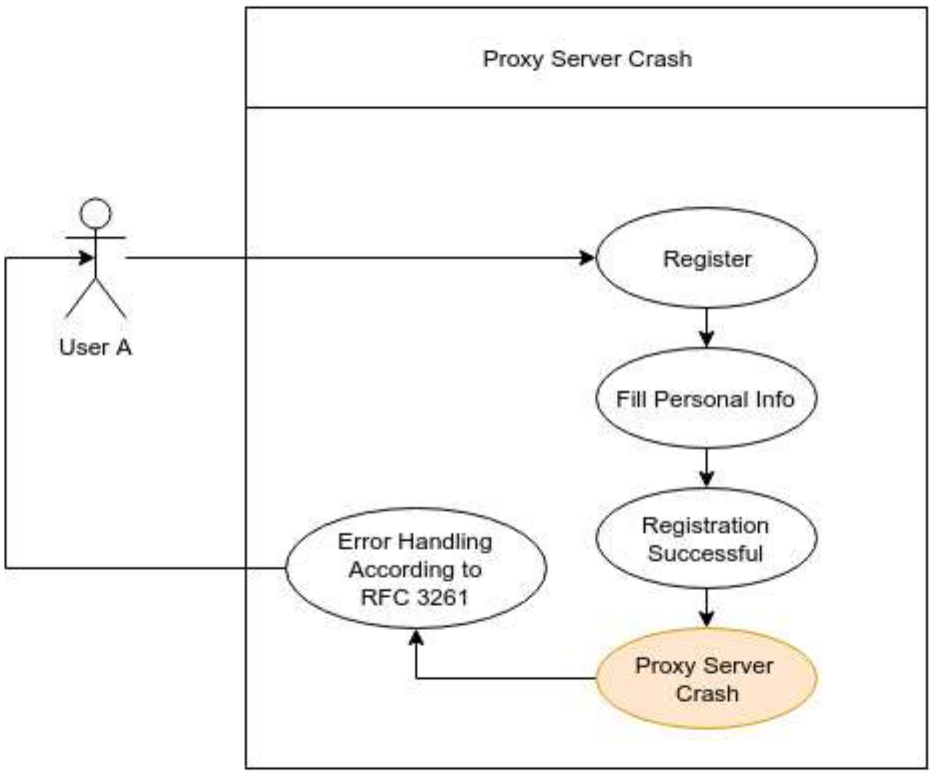


Image 9: Proxy Server Crash

Sequencing Diagrams

For each use case which is mentioned in the project description we create a sequencing diagram.

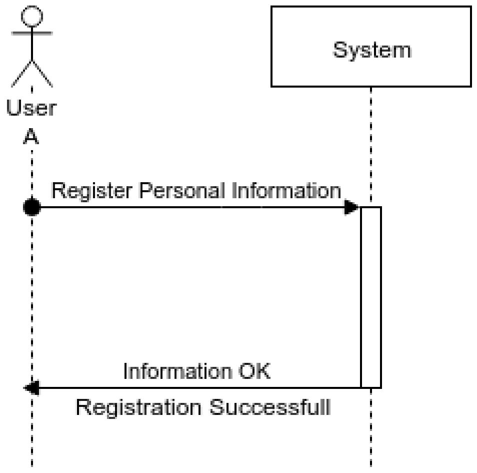


Image 10: Register

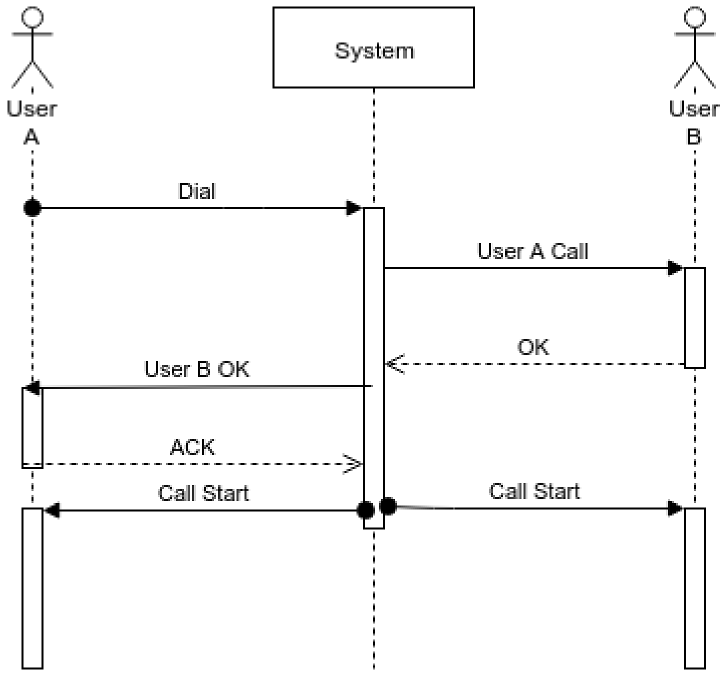


Image 11: Normal Call

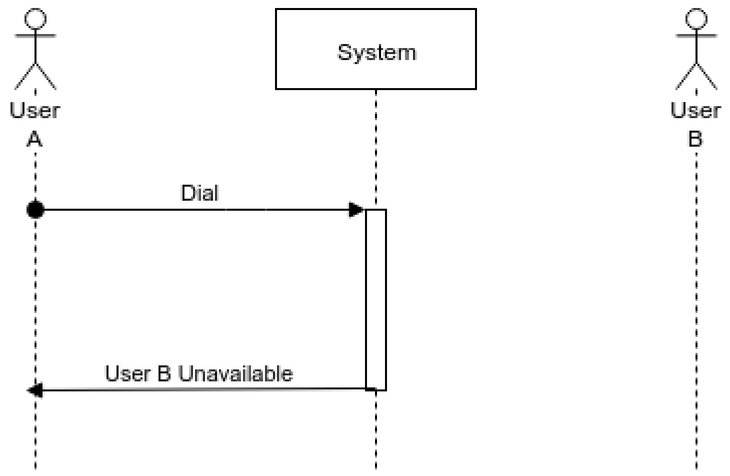


Image 12: Call blocking

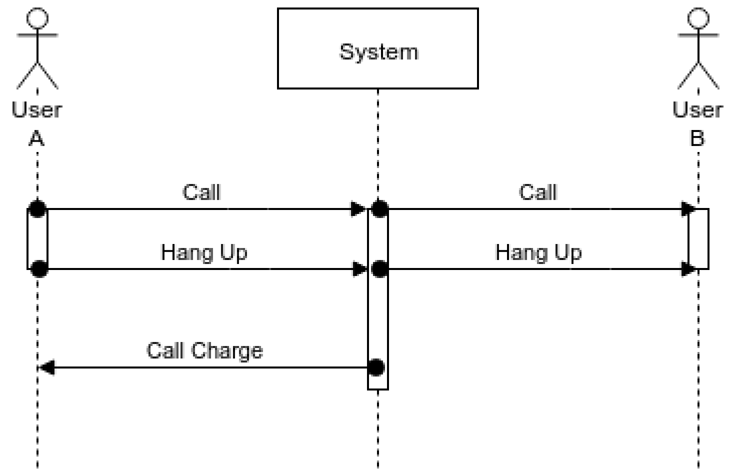


Image 13: Call charging

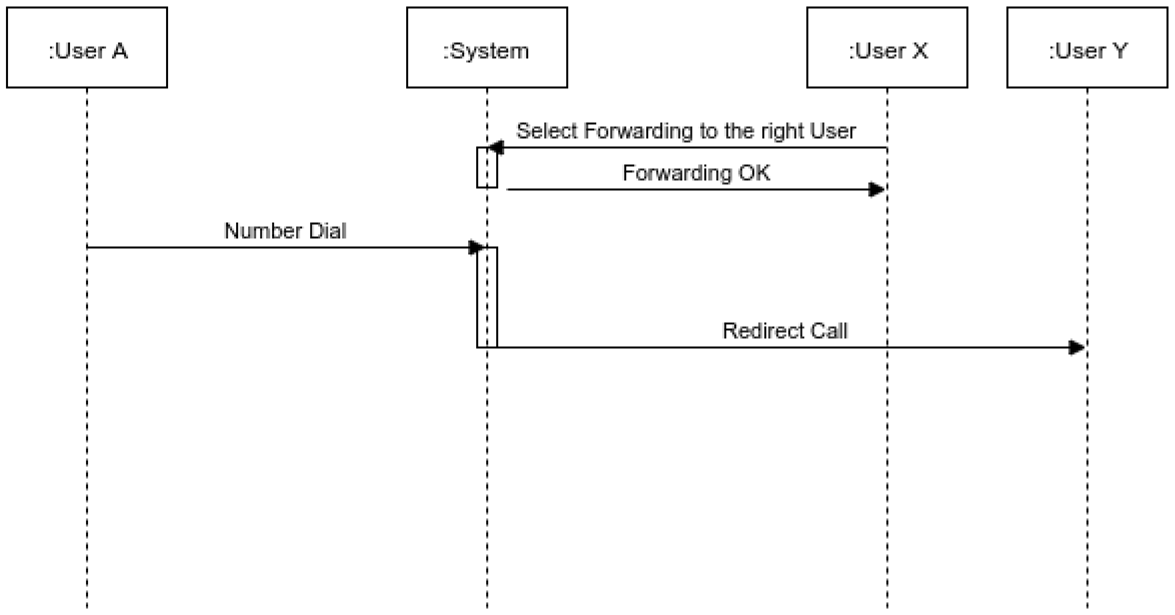


Image 14: Simple forwarding

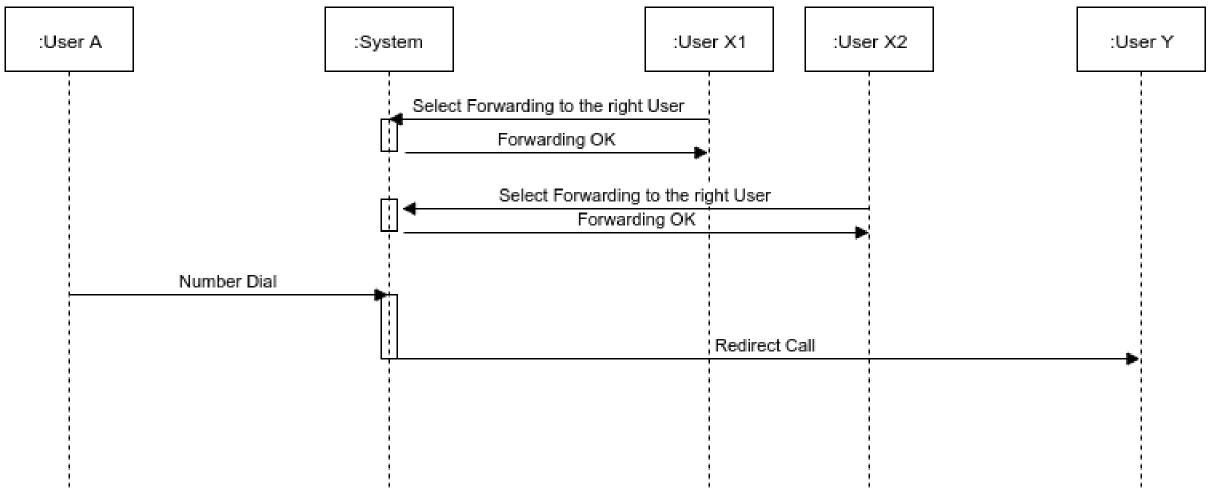


Image 15: Multiple forwarding

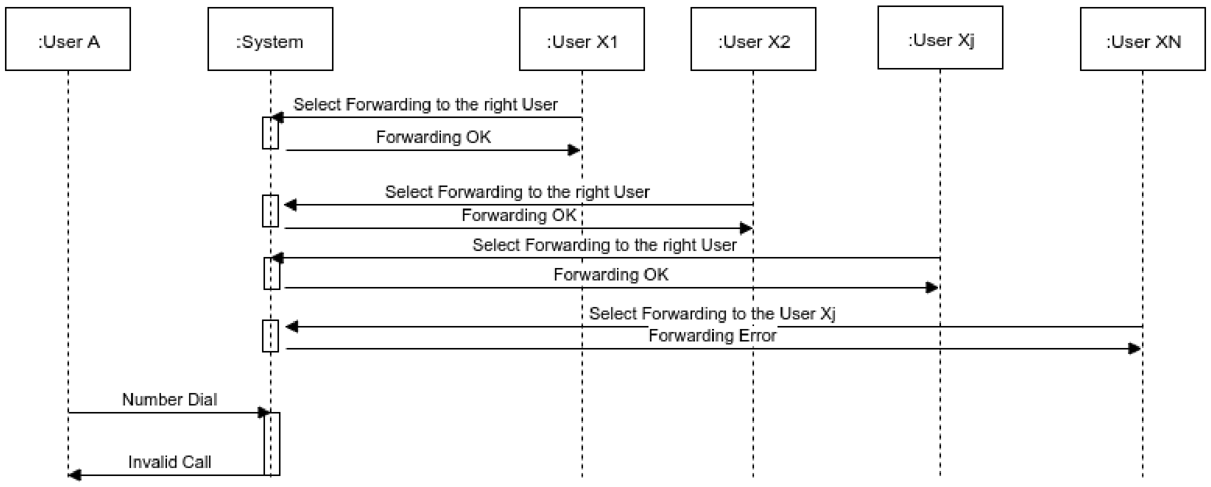

:User A
:System
:User X1
:User X2
:User Xj
:User XN
Select Forwarding to the right User
Forwarding OK
Select Forwarding to the right User
Forwarding OK
Select Forwarding to the right User
Forwarding OK
Select Forwarding to the User Xj
Forwarding Error
Number Dial
Invalid Call

Image 16: General circular invalid forwarding

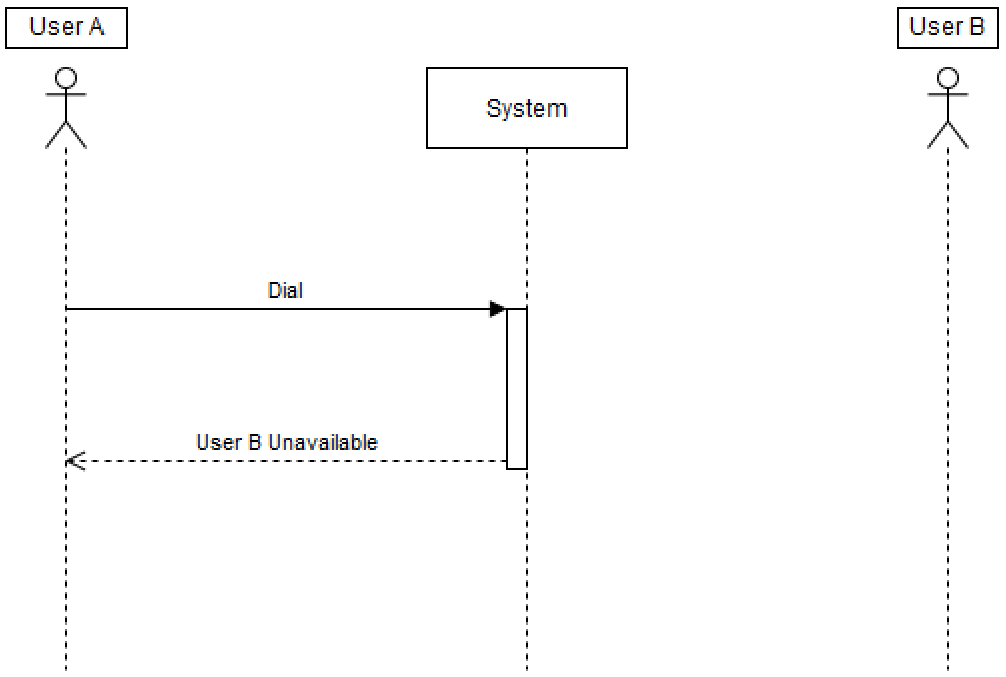


Image 17: User B not connected

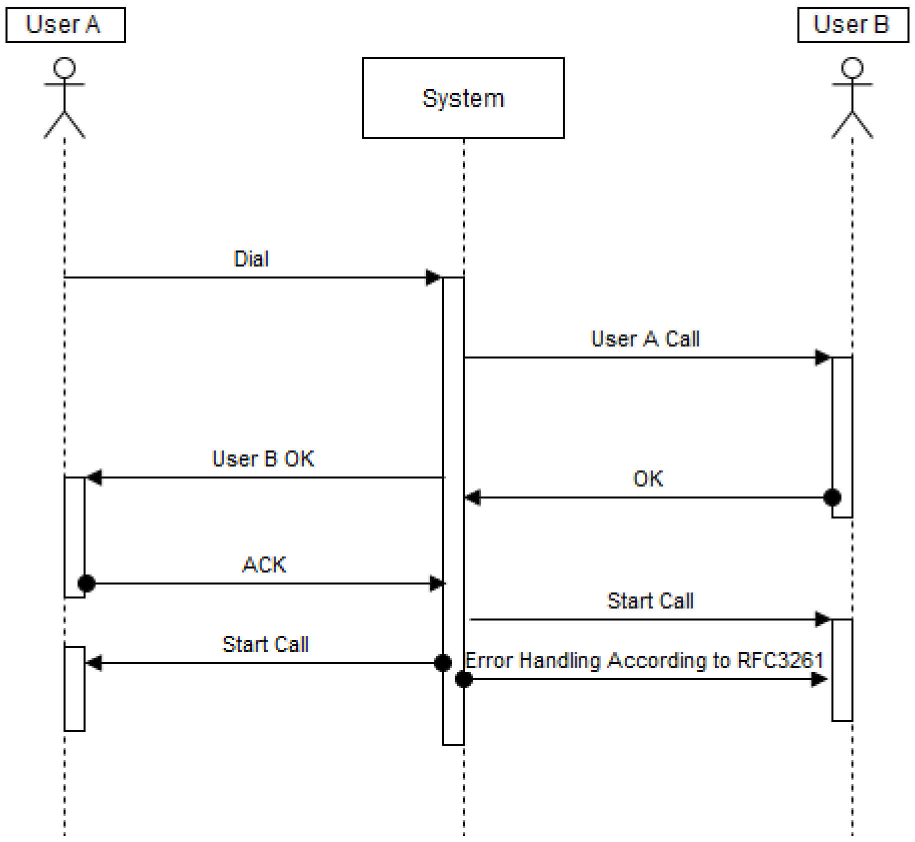


Image 18: User A SIP crashed

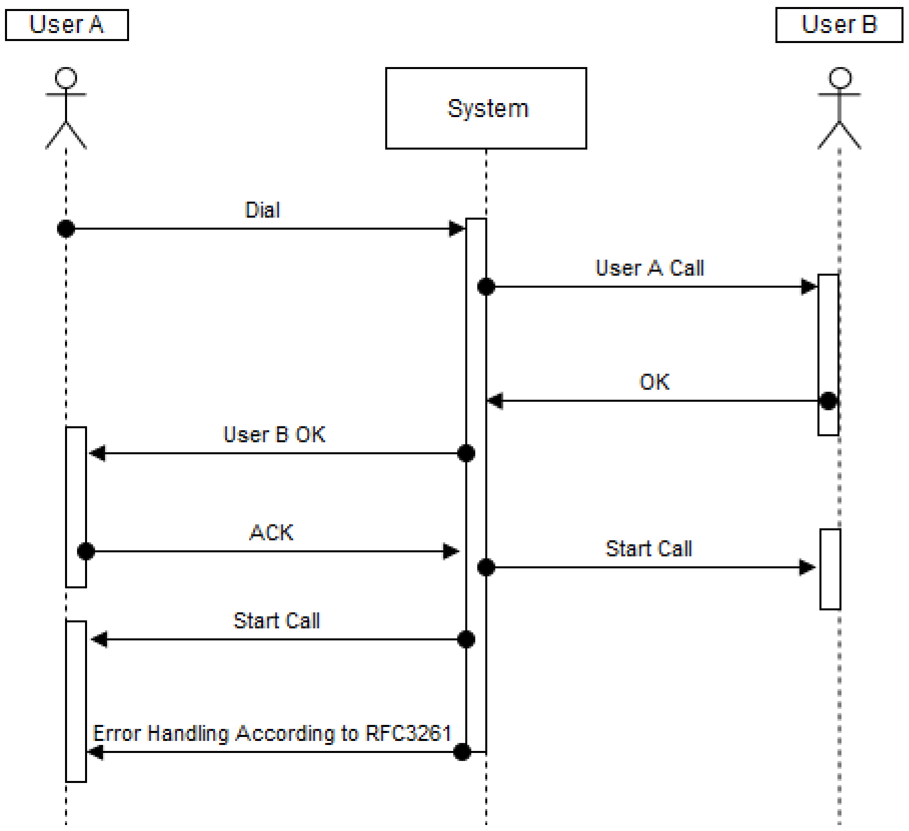


Image 19: User B SIP crashed

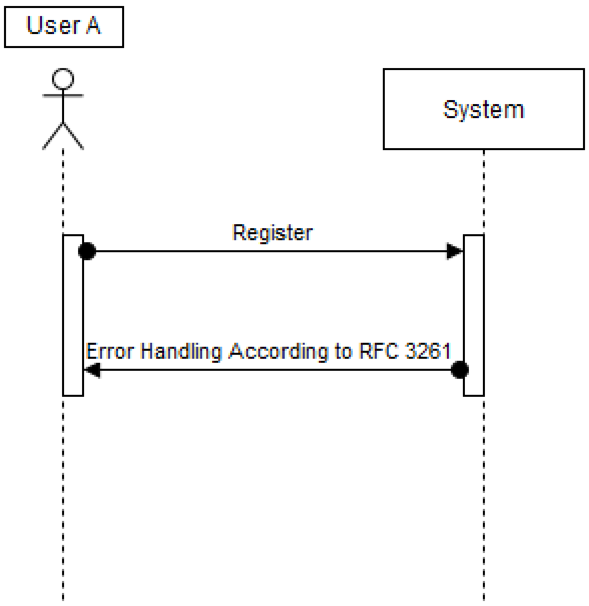


Image 20: Proxy server crashed

Domain Class Diagrams

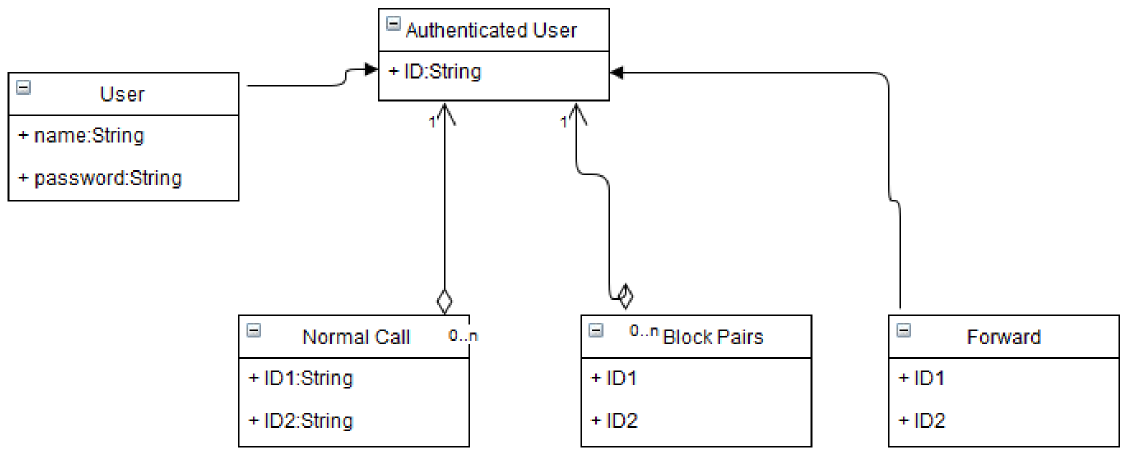


Image 21: User View

<User>

Description	It's the entity that keeps information for each person who registers and wants to be authenticated in order to use the system.
Attributes	• name • password
Responsibilities	It keeps information for each person.
Business Rules	The system needs to authenticate this entity in order for the second to use the system's services.

<Authenticated User>

Description	It's the entity that keeps information for the people who have been authenticated by the system.
Attributes	• ID
Responsibilities	It keeps the ID of each person so he can perform several actions.
Business Rules	Authenticated users are given an ID which is being added in the system's authenticated signed in users.

<Normal Call>

Description	Two authenticated users may communicate with each other.
Attributes	• ID1 • ID2
Responsibilities	The application used for the communication performs the call from ID1 to ID2.
Business Rules	Only authenticated users can perform a call.

<Block Pairs>

Description	An authenticated user may block another user.
Attributes	• ID1 • ID2
Responsibilities	User with ID1 wants to block user with ID2.
Business Rules	Only authenticated users can block other users, also authenticated.

<Forward Call>

Description	An authenticated user wants to forward a call to another authenticated user.
Attributes	• ID1 • ID2
Responsibilities	User with ID1 forwards a call to user with ID2.
Business Rules	Only authenticated users can forward calls.

Collaboration Diagrams

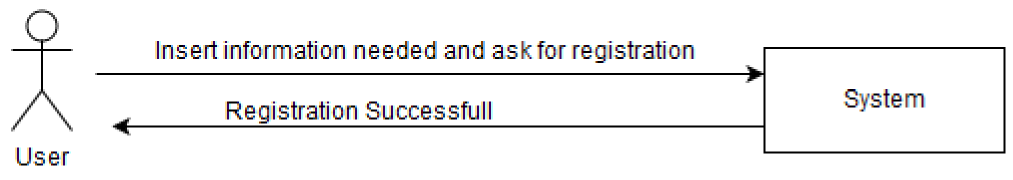


Image 22: Register

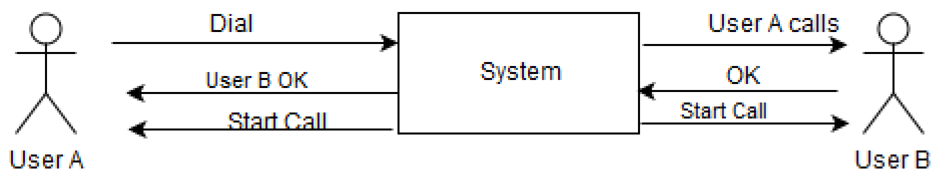


Image 23: Normal Call

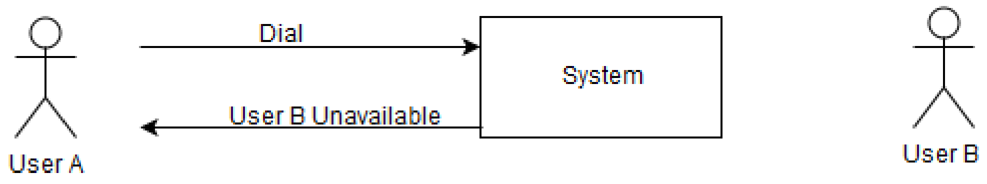


Image 24: Call blocking

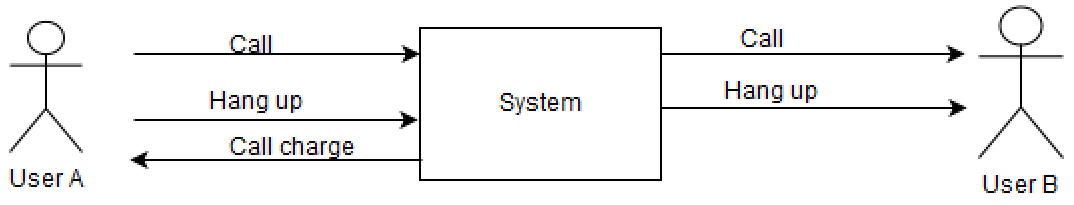


Image 25: Call charge

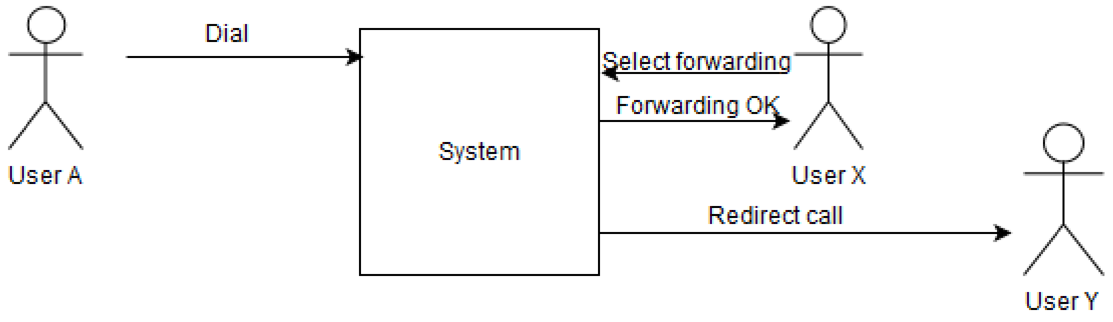


Image 26: Simple forwarding

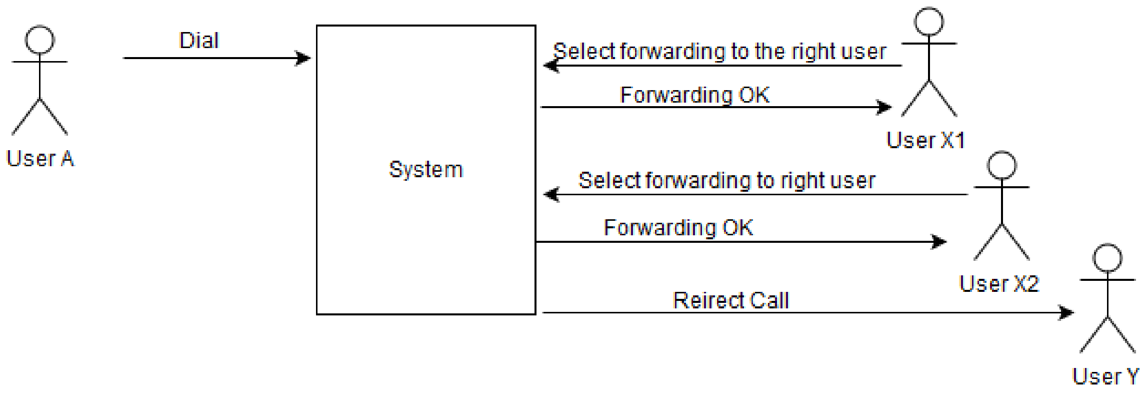


Image 27: Multiple forwarding

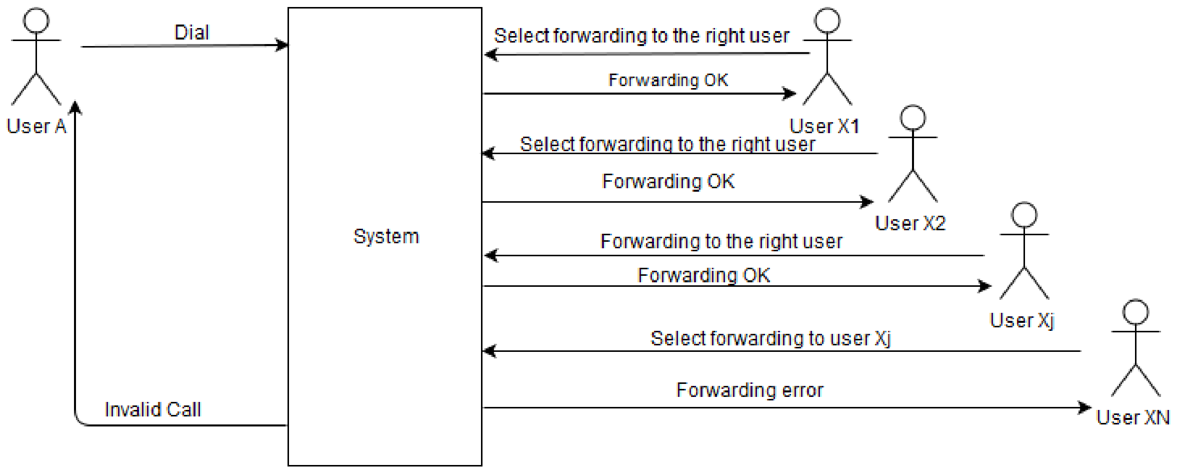

User A
Dial
System
Select forwarding to the right user
Forwarding OK
User X1
Select forwarding to the right user
Forwarding OK
User X2
Forwarding to the right user
Forwarding OK
User Xj
Select forwarding to user Xj
Forwarding error
User XN
Invalid Call

Image 28: General circular invalid forwarding

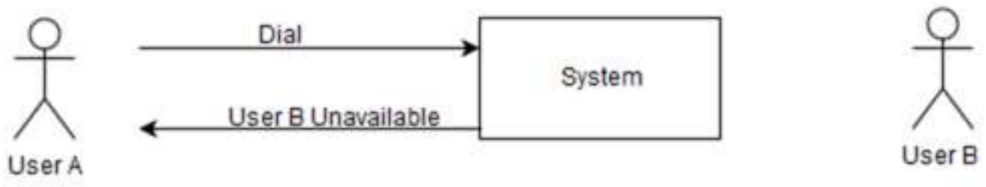


Image 29: User B not connected

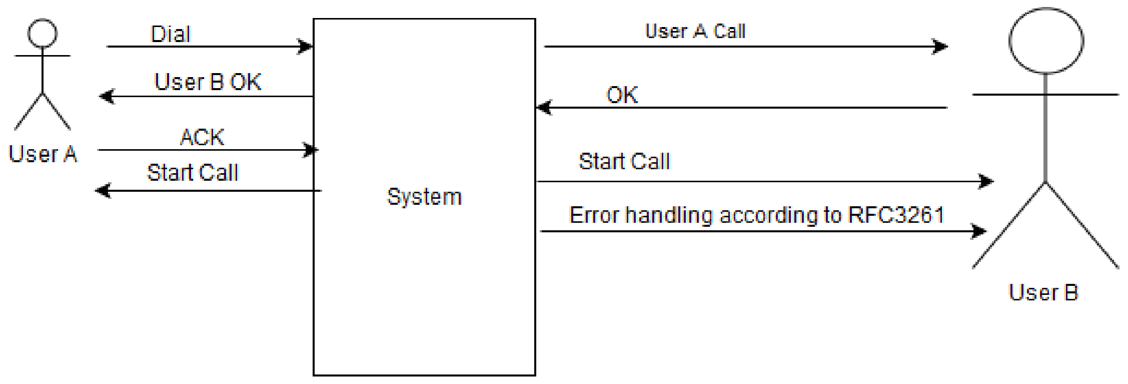


Image 30: User A SIP crashed

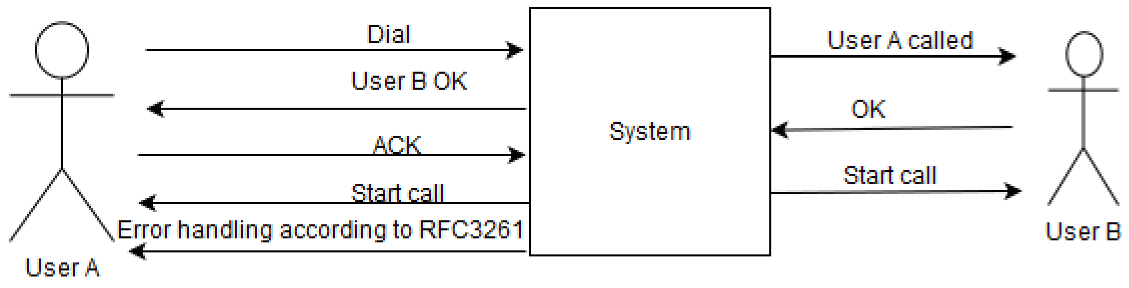


Image 31: User B crashed

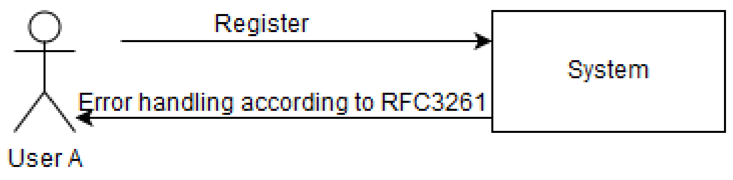


Image 32: Proxy server crashed