IT2020 – Software Engineering
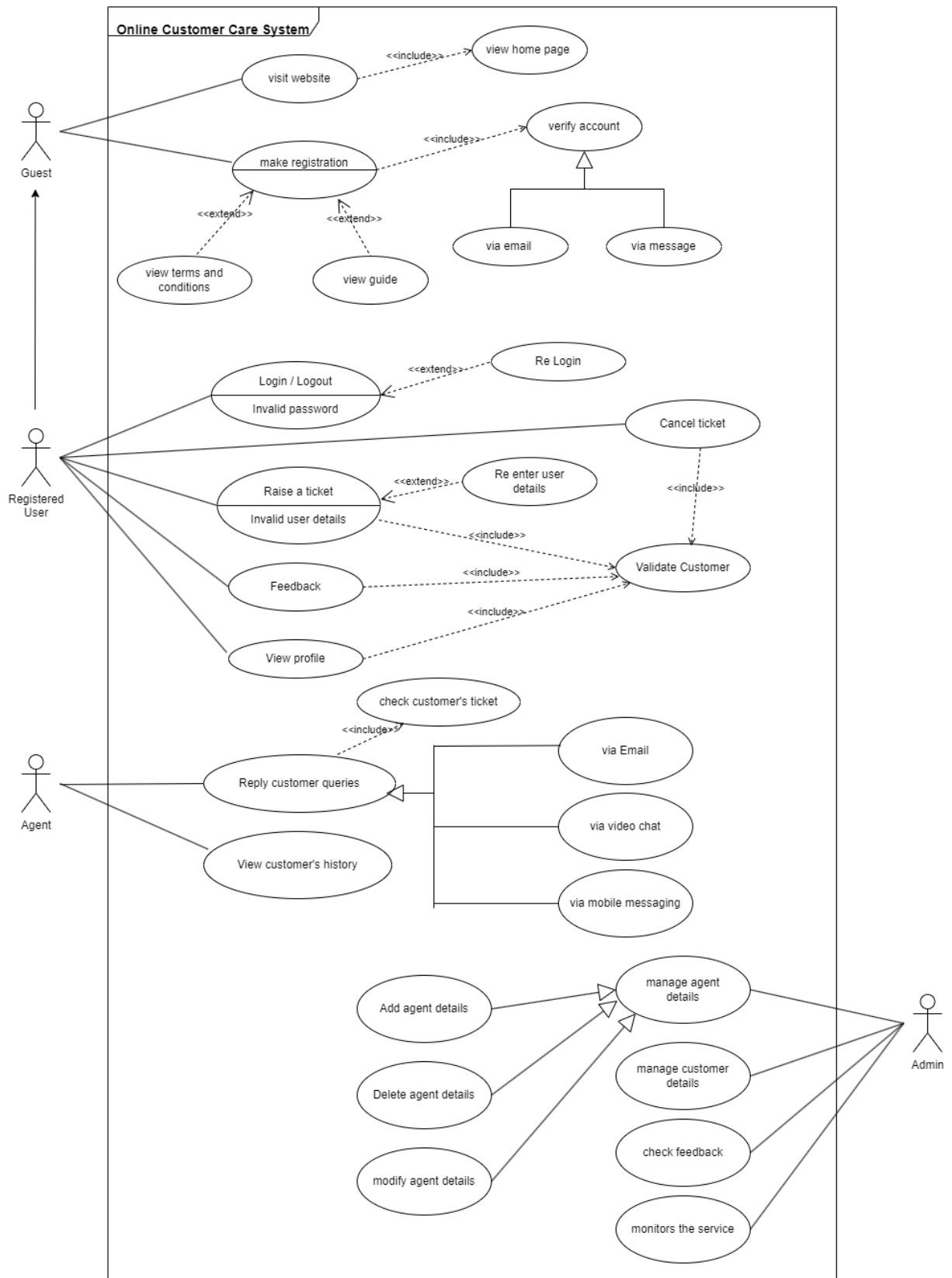# Year 2, Semester II, 2022
Group Assignment

Online customer care system

Group Details

Batch: Y1.S1

SE/OOP Group Number: MLB_WD_2022_S2_134

	Student Registration Number	Student Name
1	IT21254420	Dewpura D.D.W.C
2	IT21340628	Abeywickrama A.A.D
3	IT21212772	Koggala D.B.K.D
4	IT21240324	Kulasuriyalage H.K.K

Online Customer Care System

Actors: Guest, Registered User, Agent, Admin

Guest:
- visit website <<include>> view home page
- make registration <<include>> verify account
  - verify account: via email, via message
  - make registration <<extend>> view terms and conditions
  - make registration <<extend>> view guide

Registered User:
- Login / Logout (Invalid password) <<extend>> Re Login
- Cancel ticket <<include>> Validate Customer
- Raise a ticket (Invalid user details) <<extend>> Re enter user details
- Raise a ticket <<include>> Validate Customer
- Feedback <<include>> Validate Customer
- View profile <<include>> Validate Customer

Agent:
- Reply customer queries <<include>> check customer's ticket
  - Reply customer queries: via Email, via video chat, via mobile messaging
- View customer's history

Admin:
- manage agent details: Add agent details, Delete agent details, modify agent details
- manage customer details
- check feedback
- monitors the service

2

Number	01
Name	Make registration
Summary	Guest user register to the system.
Priority	5
Preconditions	Entered to the signup page.
Postconditions	Registered to the system and make a user account.
Primary Actor(s)	Guest user
Trigger	User has chosen to register to the system.

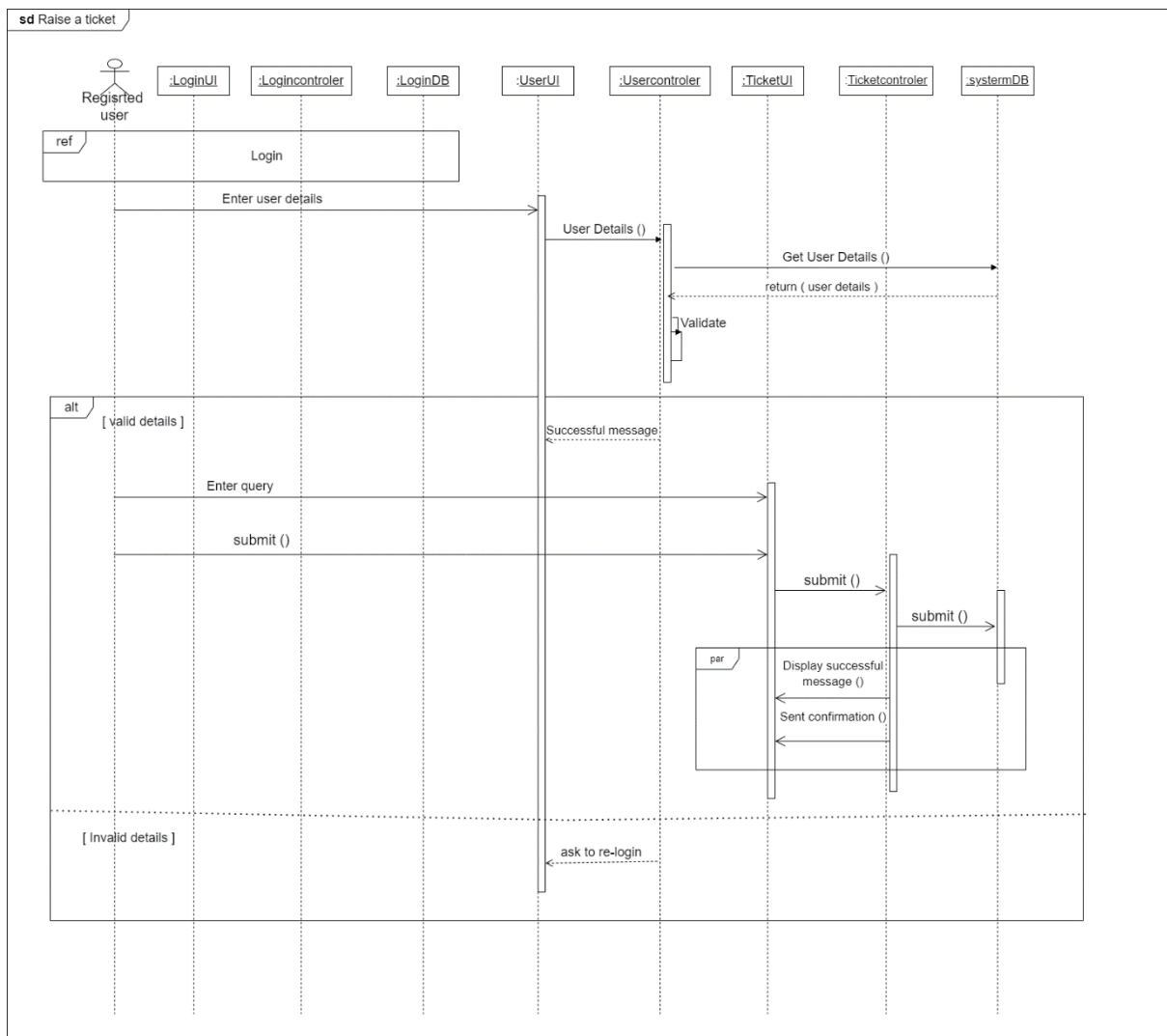| Main Scenario	Step	Action
	1	User visits the website and view homepage.
	2	User clicks the registration page.
	3	System displays registration form.
	4	User enters username and user details.
	5	System verifies user details and username.
	6	User enters a password.
	7	System validates the password.
	8	User enters password confirmation.
	9	System verifies the password confirmation.
	10	User agrees to terms and conditions and submits the form.
	11	System sends a verification code to user via email or via message.
	12	User enters the verification code.
	13	System verifies the code.
	14	System creates a user account and store user details.
	15	System shows successfully registered message.
Extensions	**Step**	**Branching Action**
	5a	System notifies username is already exists, asks to re-enter.
	7a	System notifies password is invalid, asks to re-enter.
	9a	System notifies that the two passwords does not match, asks to re-enter.
	10a	If user doesn't accept terms and conditions, system displays "Please accept the terms and conditions to continue registration" message.
	13a	System notifies verification code is not valid, asks to re-enter.

Sequence diagram - Make registration
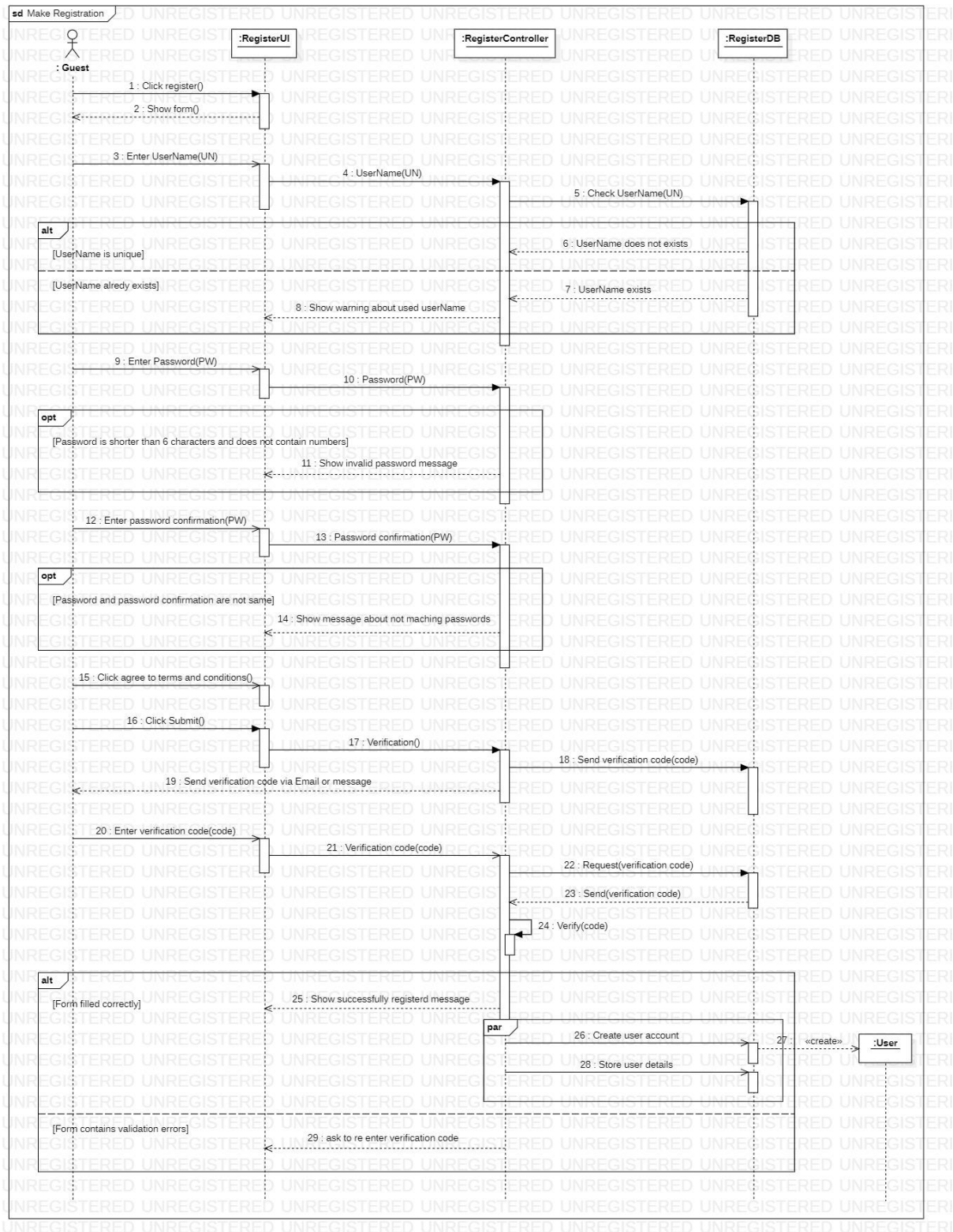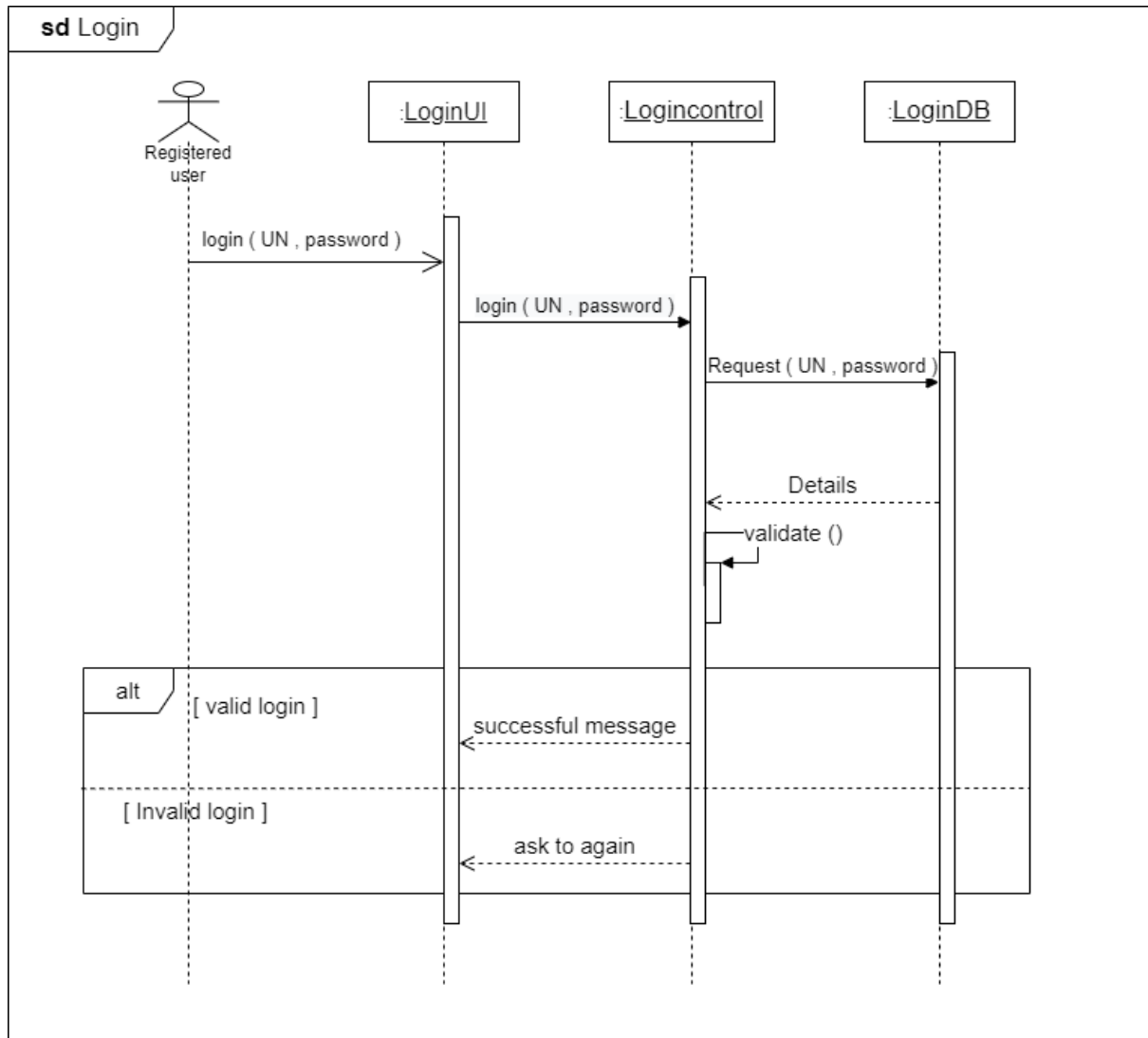
Communication diagram - Make registration

Interaction Make registration

6.1.2:Send verification code via email or message

3.0:Enter password()
2.0:Enter UserNam(UN)
1.0:Click register()

Guest

5.0:click agree to terms and conditions
4.0:Enter password confirmation()

7.0:Enter verification code(code)
6.0:Click submit()

1.0.1:Show form

:RegisterUI

4.1:Password confirmation()
3.1:Password()
2.1:UserName(UN)

4.1.1[Passwords are not same]:Show message about not matching passwords

7.1.4 [form filled correctly]:Show successfully registered message

7.1.5:[Form contains validation errors]:Ask to re-enter verification code

7.1:Verification code(code)
6.1:Verification()

2.2.3:Show warning about used userName

3.1.1[Password is short]:Show invalid password message

:RegisterController

7.1.3:Verify(code)

2.2:Check UserName(UN)

6.1.1:Send verification code(code)

7.1.2:Send(code)

7.1.1:Request(verification code)

2.2.2[UserName already exists]:UserName exists

2.2.1[UserName is unique] UserName doesn't exists

7.2A:Create user account

7.2B:Store user details

7.2.1 <<create>>

:User

:RegisterDB

Use case scenario – Raise a ticket

Number	02	
Name	Raise a ticket	
Summary	Registered user raises a ticket using system	
Priority	5	
Preconditions	Entered to the login page.	
Postconditions	Raise a ticket using system.	
Primary Actor(s)	Registered user	
Trigger	User has chosen to raise a ticket using system.	
Main Scenario	Step	Action
	1	System display login page.
	2	User enters username, password.
	3	System validate login.
	4	User Enter user details.
	5	System validate user details.
	6	User enter query.
	7	User submit query.
	8	System display successful message and sent confirmation.
Extensions	Step	Branching Action
	3a	If user login details invalid, system asks to re-login to the system
	5a	If user details Invalid, system ask to re-enter details.

Sequence diagram – Raise a ticket

sd Raise a ticket

Registered user | :LoginUI | :Logincontroler | :LoginDB | :UserUI | :Usercontroler | :TicketUI | :Ticketcontroler | :systemDB

ref — Login

Enter user details

User Details ()

Get User Details ()

return (user details)

Validate

alt

[valid details]

Successful message

Enter query

submit ()

submit ()

submit ()

par

Display successful message ()

Sent confirmation ()

[Invalid details]

ask to re-login

Sequence diagram – User login

sd Login

Registered user → :LoginUI : login (UN , password)

:LoginUI → :Logincontrol : login (UN , password)

:Logincontrol → :LoginDB : Request (UN , password)

:LoginDB --> :Logincontrol : Details

:Logincontrol : validate ()

alt
[valid login]
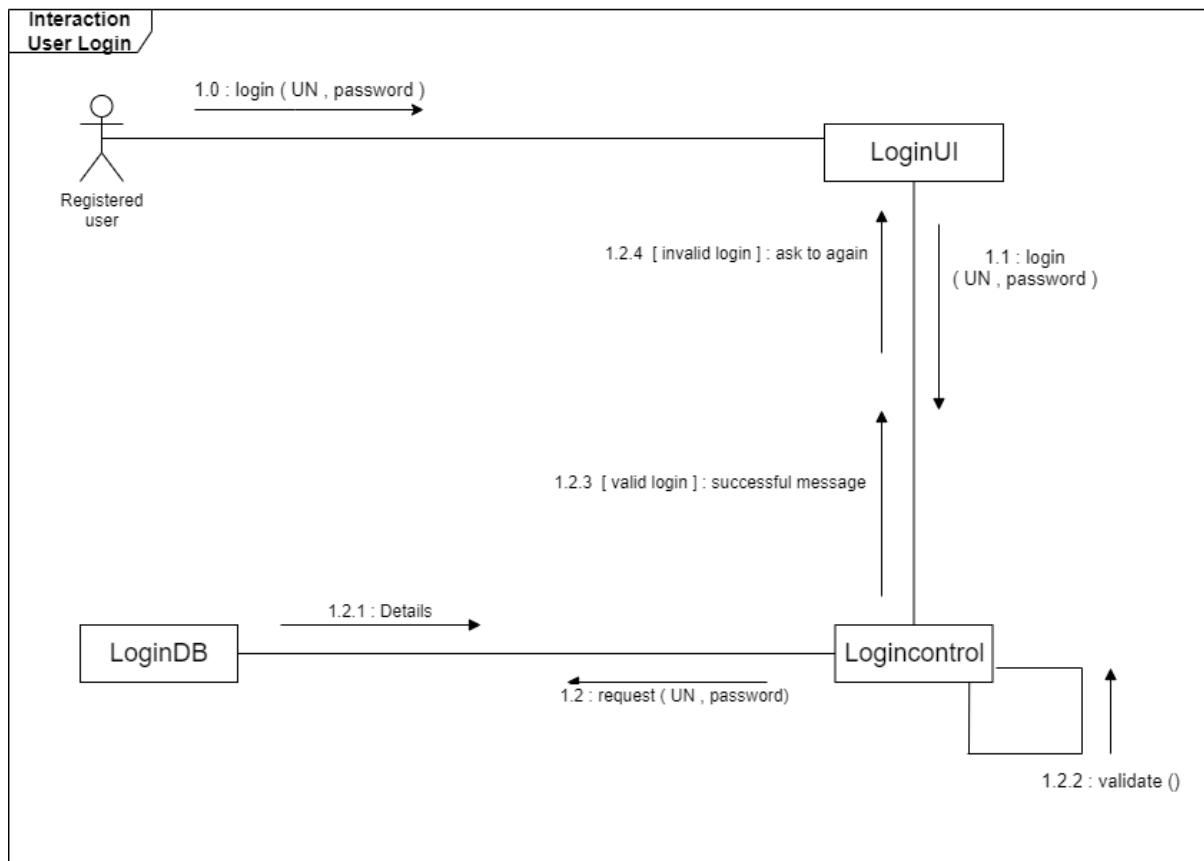:Logincontrol --> :LoginUI : successful message

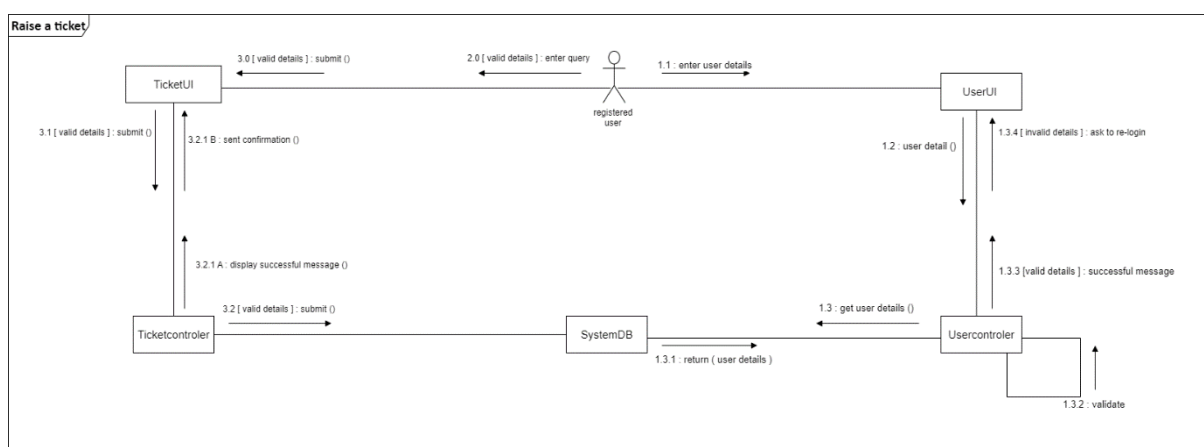[Invalid login]
:Logincontrol --> :LoginUI : ask to again

Communication diagram – User login

Communication diagram – Raise a ticket

Use case scenario – Reply to customer queries

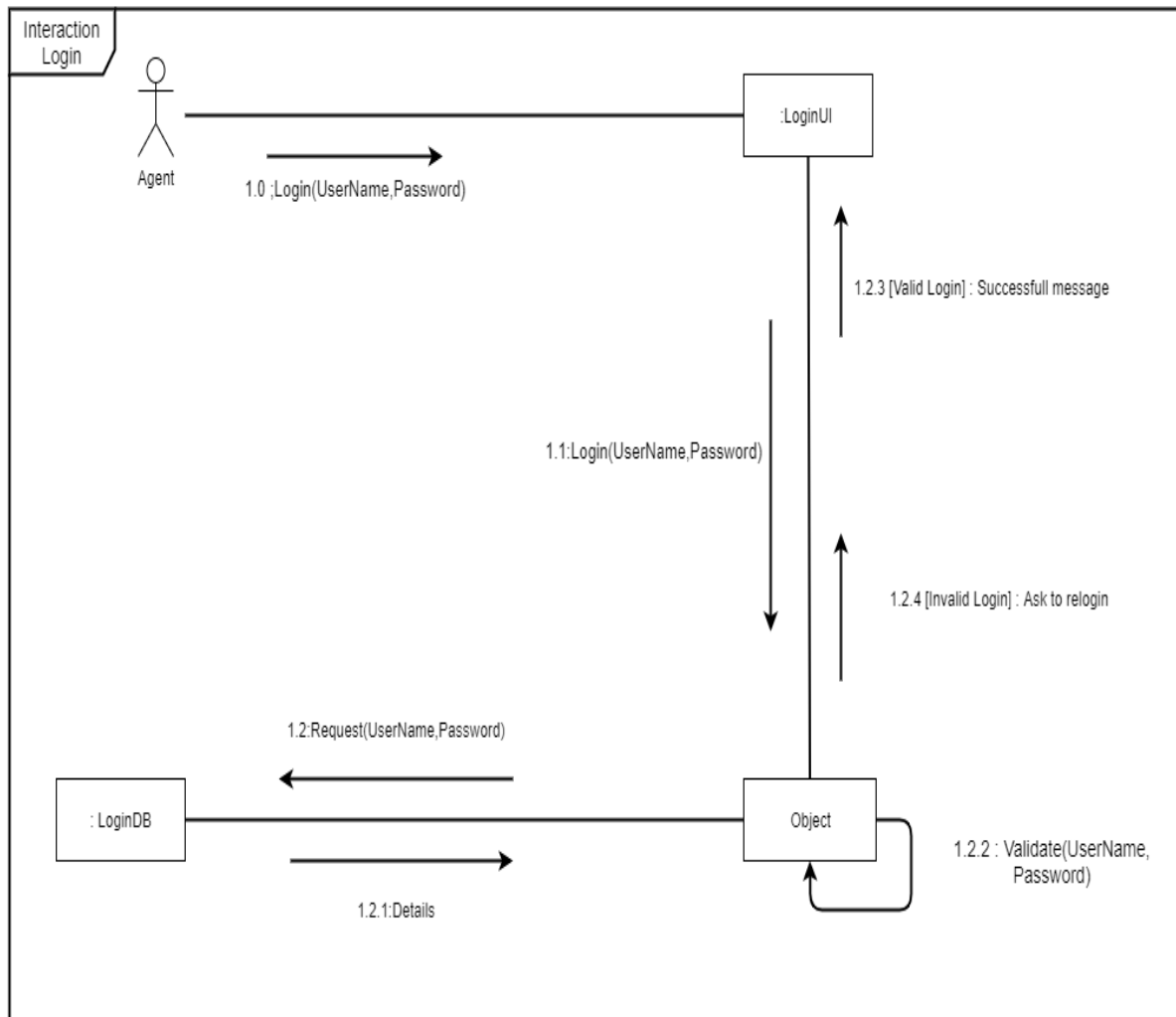| Number	03	
Name	Reply to customer queries	
Summary	Agent replies to customers' queries.	
Priority	5	
Preconditions	Entered to the query page.	
Postconditions	Check customer queries and reply to queries.	
Primary Actor(s)	Agent	
Trigger	Agent has chosen to reply to customer queries.	
Main Scenario	**Step**	**Action**
	1	System display login page.
	2	Agent enters username, password.
	3	System validates login
	4	Agent clicks the check queries.
	5	System displays queries page.
	6	System asks to select priority of the queries.
	7	Agent selects priority.
	8	System displays queries according to the selected priority.
	9	Agent checks queries and add solutions.
	10	System displays solution saved message.
	11	System store details and send an e-mail to user.
Extensions	**Step**	**Branching Action**
	3a	If user login details invalid, system asks to re-login to the system.
	11a	System notifies user details are invalid, asks to re-enter user details.

Sequence diagram - Reply to customer queries
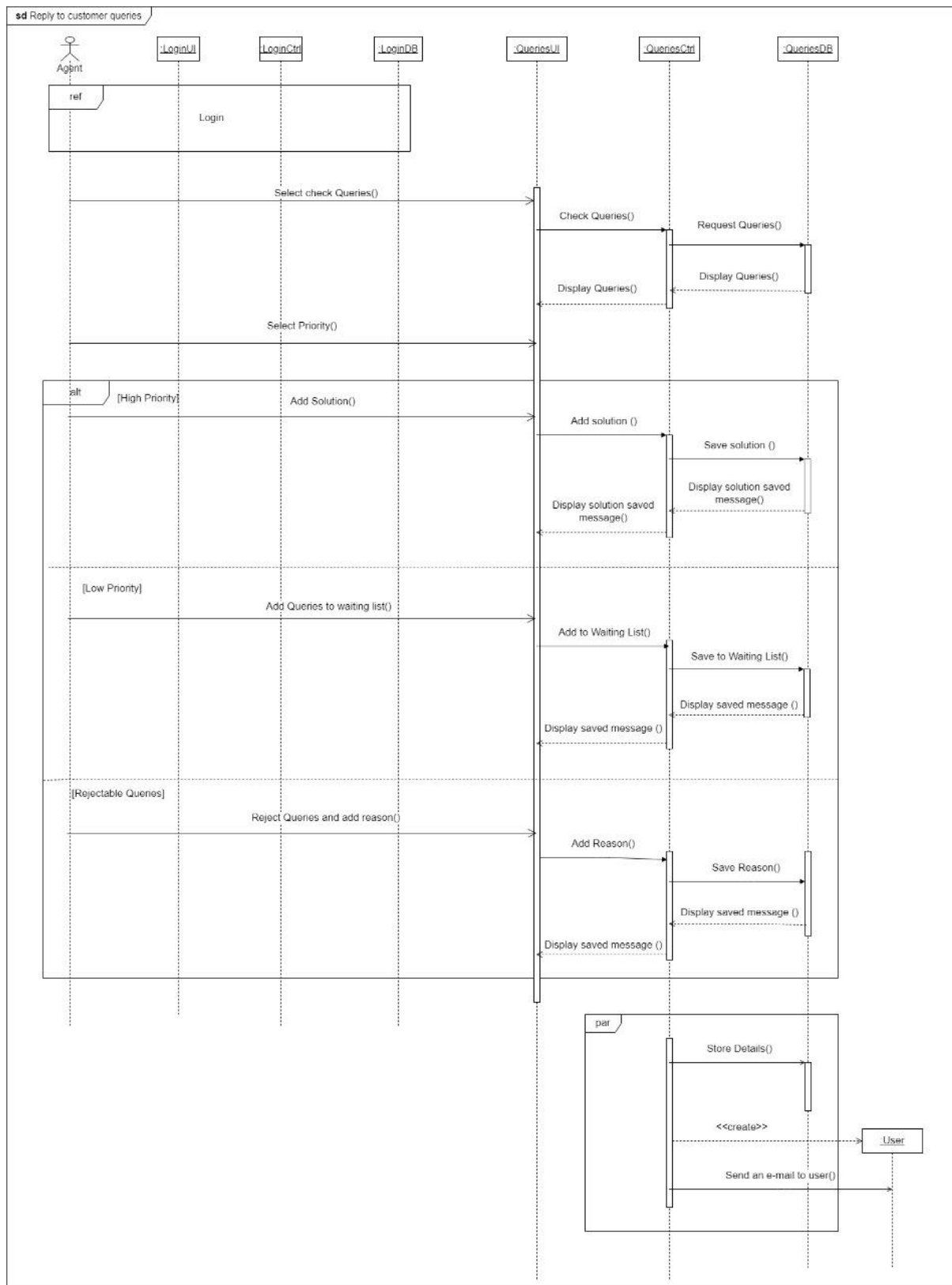
Sequence diagram – Agent login

sd Login

```
        Agent            :LoginUI        :LoginCountrol         :LoginDB

Login(UserName,Password)
         ──────────────────>│
                            │  Login(UserName,Password)
                            │────────────────────────>│
                            │                          │  Request(UserName,Password)
                            │                          │─────────────────────────>│
                            │                          │                           │
                            │                          │          Details          │
                            │                          │<─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
                            │                          │                           │
                            │                          │┐ Validate(UserName,Password)
                            │                          ││                           │
                            │                          │◄┘                          │
```

alt

[Valid Login]

Successfull message
`<─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─`

- -

[Invalid Login]

Ask to ReLogin
`<─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─`
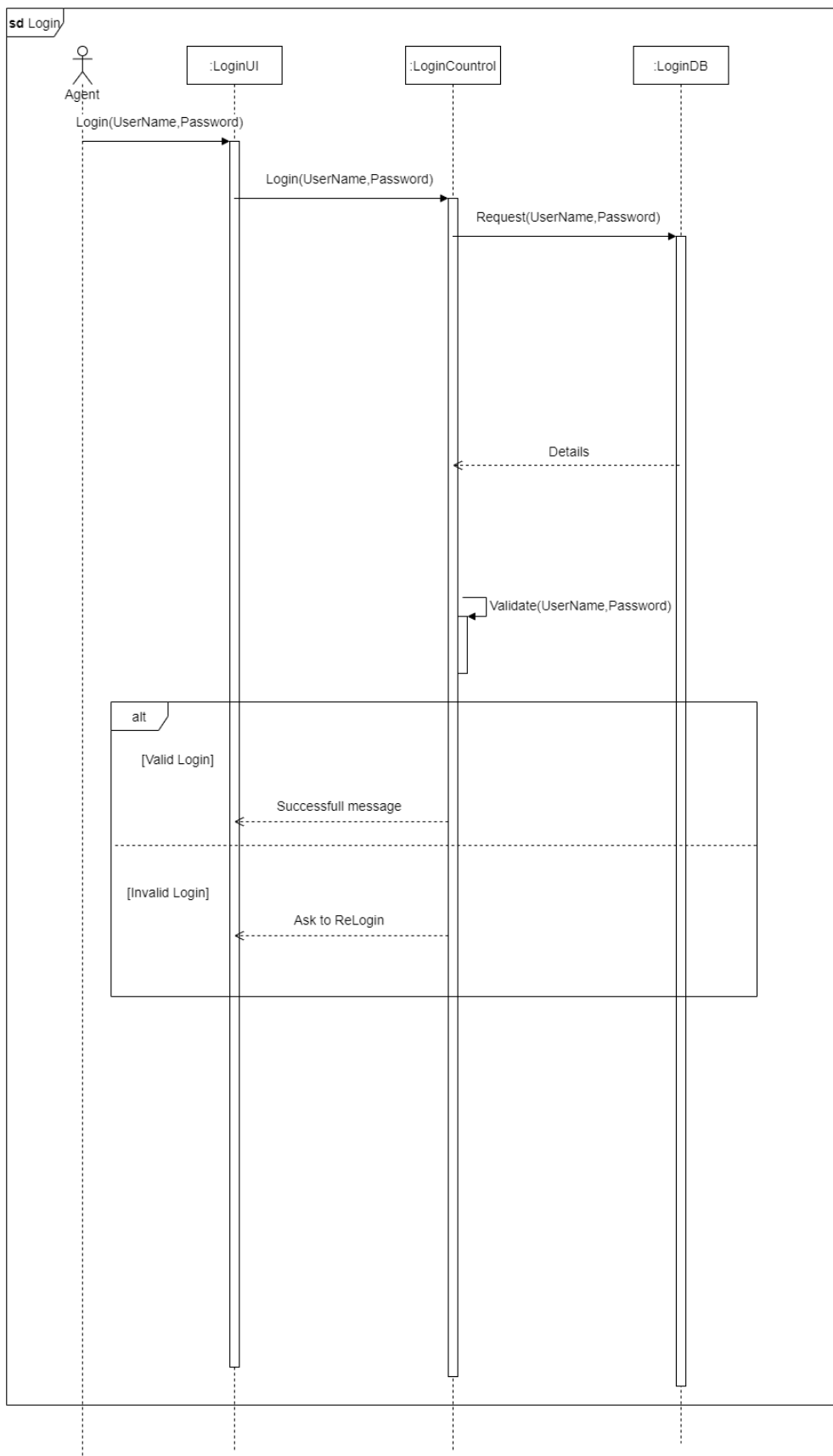
Communication diagram – Agent login

Interaction Login

Agent

1.0 ;Login(UserName,Password)

:LoginUI

1.2.3 [Valid Login] : Successfull message

1.1:Login(UserName,Password)

1.2.4 [Invalid Login] : Ask to relogin

1.2:Request(UserName,Password)

: LoginDB

1.2.1:Details

Object

1.2.2 : Validate(UserName, Password)

Communication diagram – Reply to customer queries

Interaction Reply to customer Queries

- 1.0 : Select Check Queries()
- 2.0 : Select Priority()
- 3.0 [High Priority] : Add Solution()
- 4.0 [Low Priority] : Add Queries to waiting list()
- 5.0 [Rejectable priority] : Reject Queries and Add reason()

Agent

QueriesUI

- 5.2 [Rejectable Queries] : Display Saved message()
- 1.1 : Check Queries()
- 4.2 [Low Priority] : Display Saved message()
- 3.1 [High Priority] : Add Solution()
- 3.2 [High Priority] : Display Solution Saved message()
- 4.1 [Low priority] : Add to waiting list()
- 1.2 : Display Queries()
- 5.1 [Rejectable Queries] : Add Reason()

- 5.2.1a : Store Details()
- 5.1.1 [Rejectable Queries] : Save Reason()
- 4.1.1 [Low priority] : Save to waiting list()
- 3.1.1 [High Priority] : Save Solution()
- 1.1.1 : Reqest Queries()

QueriesDB

QueriesCtrl

- 1.1.2 : Display Queries()
- 3.1.2 [High Priority] : Display Solution Saved message()
- 4.1.2 [Low Priority] : Display Saved message()
- 5.1.2 [Rejectable Queries] : Display Saved message()
- 5.2.1b : create()
- 5.2.1c : Send an email to user()

User

Use case scenario – Manage agent details

Number	04
Name	Manage agent details
Summary	Admin add, remove, modify agent details to the system
Priority	3
Preconditions	Admin has login to the system
Postconditions	User has added, delete, modify admin details, and send a confirmation message
Primary Actor(S)	Admin
Trigger	User has chosen to add , delete , modify admin details.

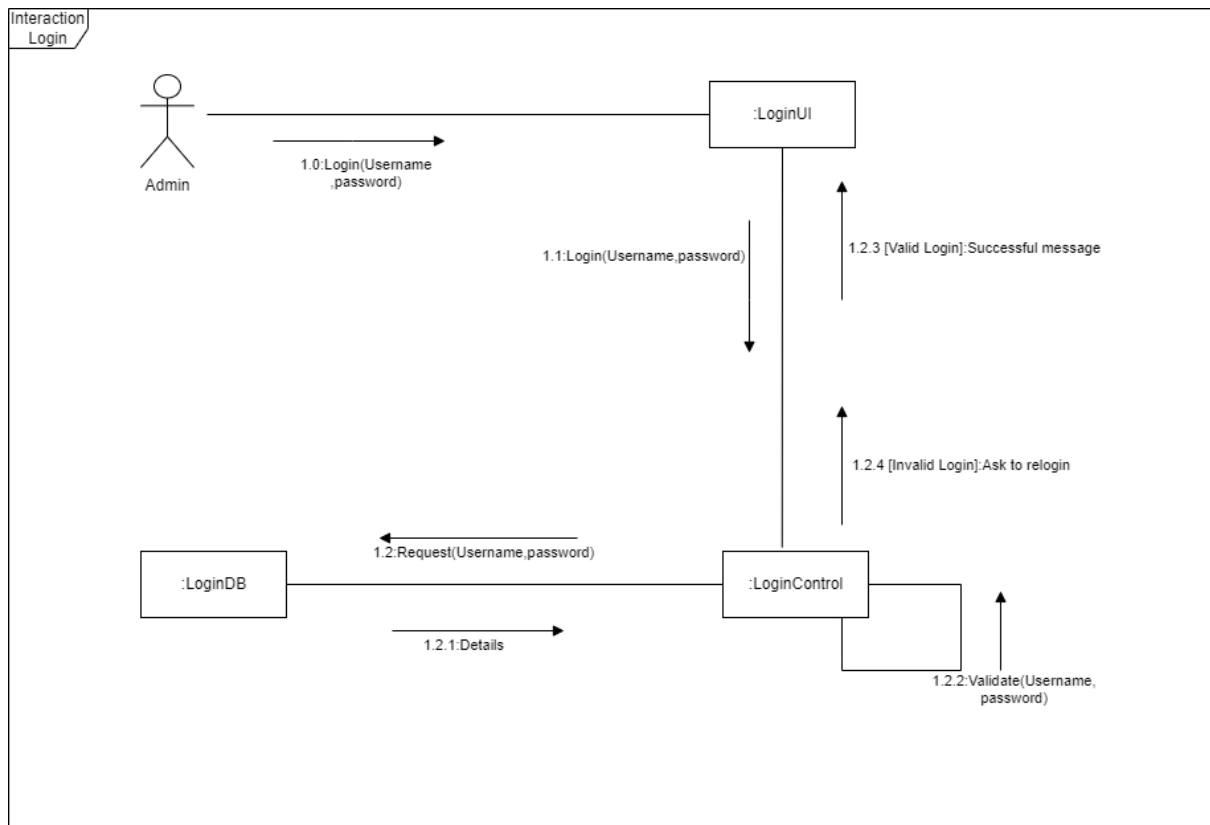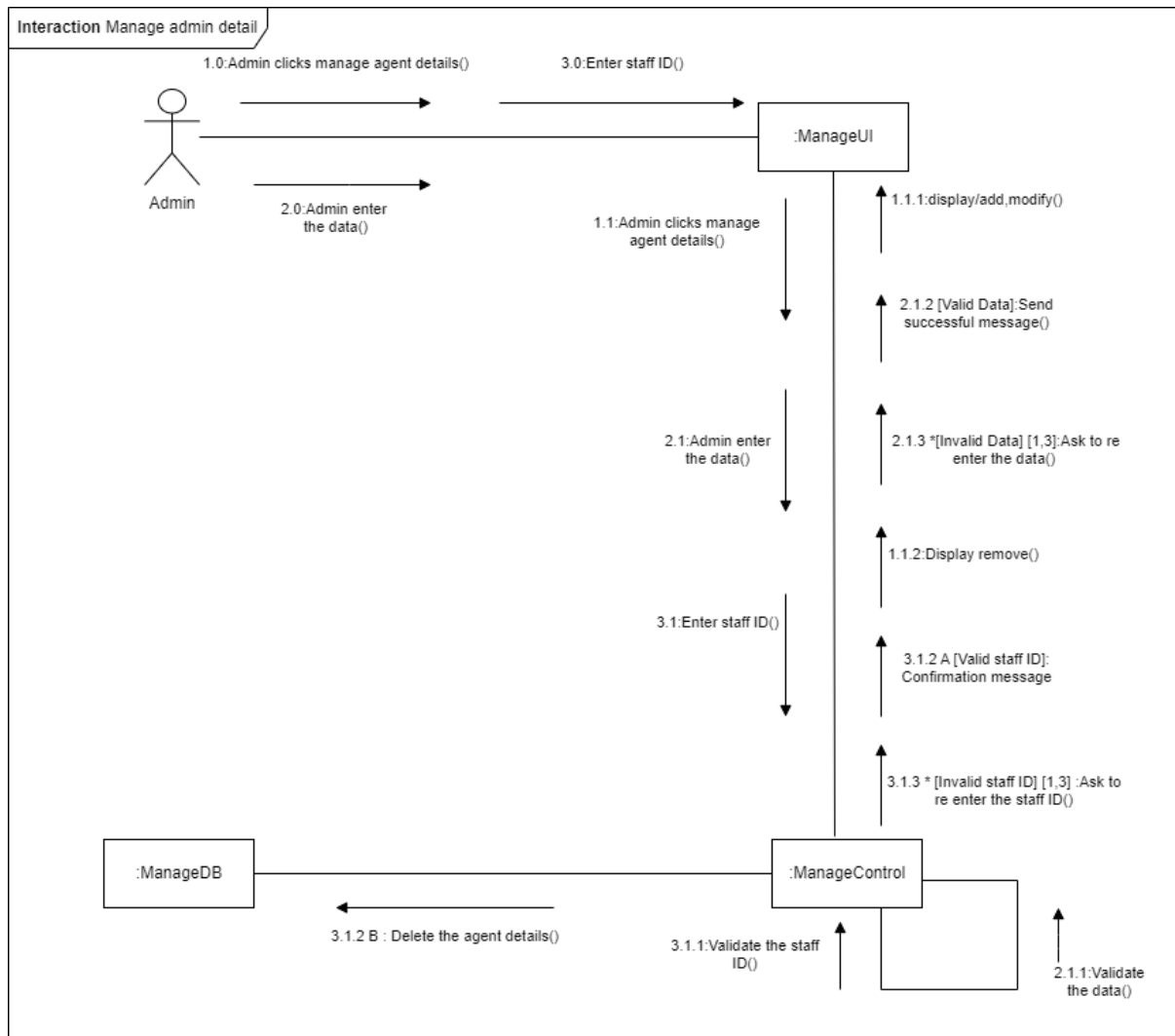| **Main Scenario**	**Step**	**Action**
	1	System shows login page to admin
	2	Admin enter username, password
	3	System validate login
	4	If the login is valid system display manage agent details
	5	Admin clicks manage agent details
	6	System display Add or modify details
	7	Admin select suitable and enter the data
	8	System validates the entered data
	9	If the data is valid, system send the successful message to admin
	10	System display Remove staff details
	11	Admin enter the agent ID
	12	System validates the agent ID
	13	If the agent ID is valid, at the same time system will remove that staff details and send a confirmation message to admin
Extensions	**Step**	**Branching Action**
	4a	Re log into the system
	9a	If the data is invalid system allow admin to reenter the data (Admin can enter the data 3 times only)
	13a	If the agent ID is not valid, system allow admin to reenter the staff ID (Admin can enter the data 3 times only)

Sequence diagram - Manage agent details

sd Manage agent details

Admin: Actor1 | :LoginUI | :LoginControl | :LoginDB | :ManageUI | :Managecontrol | :ManageDB

ref (Login)

1 : admin clicks manage agent details()

2 : Admin clicks manage agent details()

3 : display /add, modify()

4 : Admin enter the data()

5 : Admin enter the data()

6 : Validate the data()

alt

[Valid Data]

7 : Send successful message ()

[Invalid Data]

loop (1,3)

[Invalid Data]

8 : Ask to re-enter the data()

9 : Display remove()

10 : Enter staff ID()

11 : Enter staff ID()

12 : Validate the staff ID()

alt

[Valid staff ID]

par

13 : Confirmation message

14 : Delete the agent details

[Invalid staff ID]

loop (1,3)

[Invalid staff ID]

15 : Ask to re enter the staff ID()

16

Sequence diagram – Admin login
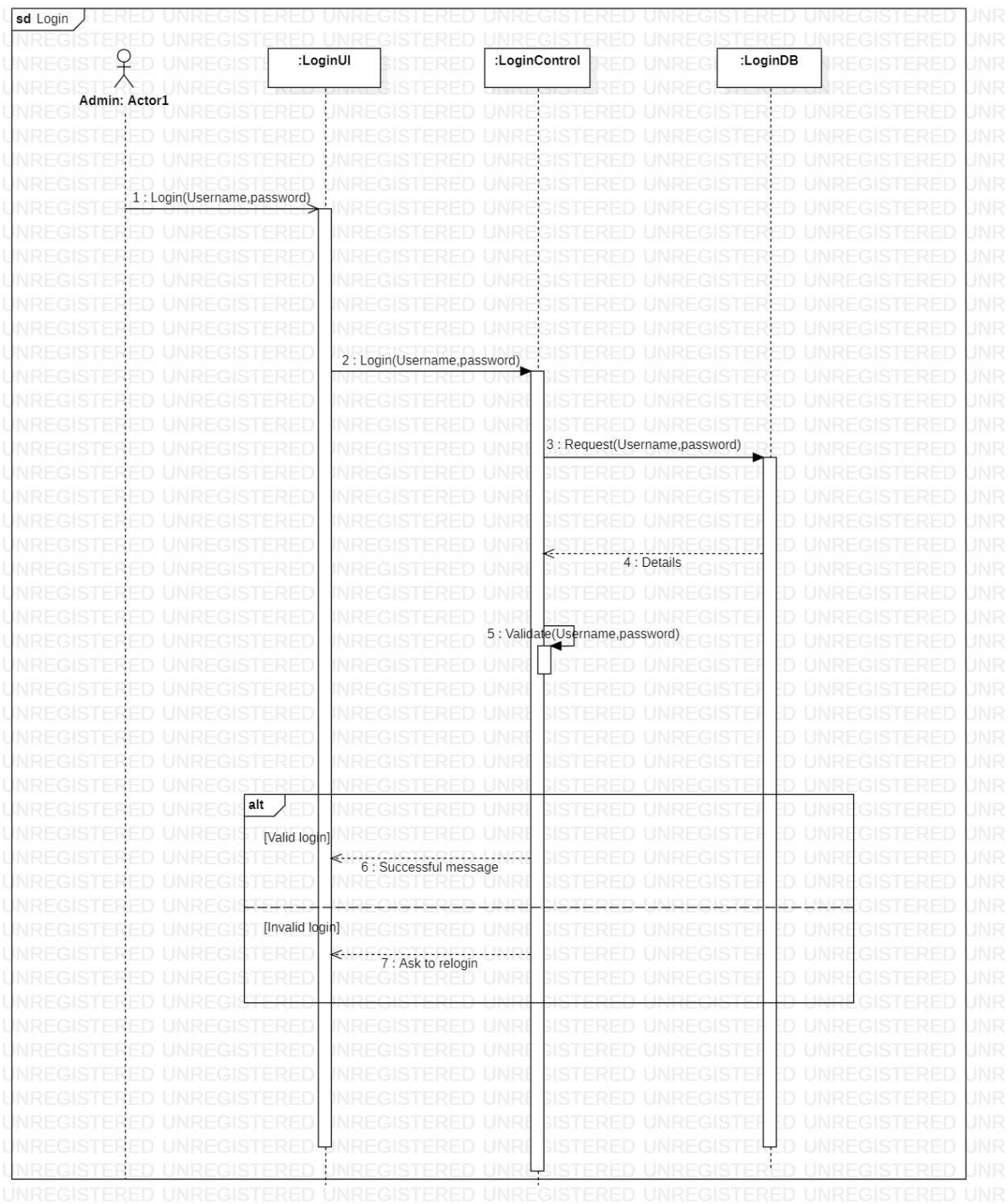
Communication diagram – Admin login

Interaction Login

Admin

1.0:Login(Username ,password)

:LoginUI

1.1:Login(Username,password)

1.2.3 [Valid Login]:Successful message

1.2.4 [Invalid Login]:Ask to relogin

1.2:Request(Username,password)

:LoginDB

:LoginControl

1.2.1:Details

1.2.2:Validate(Username, password)

Communication diagram – Manage agent details

Reg.No	Name	Function Name
IT21254420	Dewpura D.D.W.C	Make registration
IT21340628	Abeywickrama A.A.D	Raise a ticket
IT21212772	Koggala D.B.K.D	Reply to customer queries
IT21240324	Kulasuriyalage H.K.K	Manage agent details