



== FARM CITY ==

[IS 201] Object Oriented Application Development

Final Submission

Chloe Chan Li Lin – S9227750Z

Ngo Hoan Vu – G1342827T

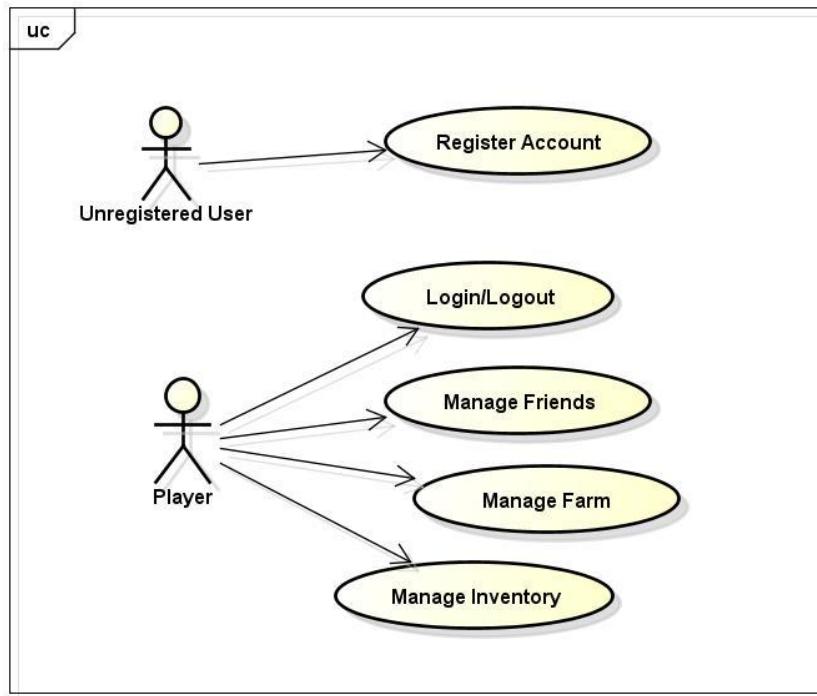
Remy Ng Zheng Yao – S9216920J

Table of Contents

1.	Use Case Diagram	3
2.	Domain Diagram	4
3.	Class Diagram	5
4.	Use Case Scenarios	7
4.1.	Use Case – REGISTER ACCOUNT.....	7
4.1.1.	Screenshots	8
4.2.	Use Case - LOGIN / LOGOUT	9
4.2.1.	Use Case Scenario: Login	9
4.2.2.	Use Case Scenario: Logout	10
4.3.	Use Case – MANAGE FRIENDS.....	11
4.3.1.	Use Case Scenario: Unfriend Friend	11
4.3.2.	Use Case Scenario: Request Friend	12
4.3.3.	Use Case Scenario: Accept Friend.....	15
4.3.4.	Use Case Scenario: Reject Friend.....	16
4.4.	Use Case – MANAGE FARM	18
4.4.1.	Use Case Scenario: Plant Crop	18
4.4.2.	Use Case Scenario: Clear Crop	20
4.4.3.	Use Case Scenario: Harvest Crop.....	22
4.5.	Use Case – MANAGE INVENTORY	24
4.5.1.	Buy Crop.....	24
4.5.2.	Gift Crop.....	26
5.	System Sequence Diagram	30
5.1.	Register.....	30
5.2.	Login / Log out	30
5.3.	Manage Friends	31
5.3.1.	Unfriend Friend	31
5.3.2.	Request Friend	31
5.3.3.	Accept Friend.....	31
5.3.4.	Reject Friend.....	32
5.4.	Manage Farm	32
5.4.1.	Plant Crop	32
5.4.2.	Clear Crop	33
5.4.3.	Harvest Crop.....	33
5.5.	Manage Inventory	33
5.5.1.	Buy Crop.....	33
5.5.2.	Gift Crop.....	34

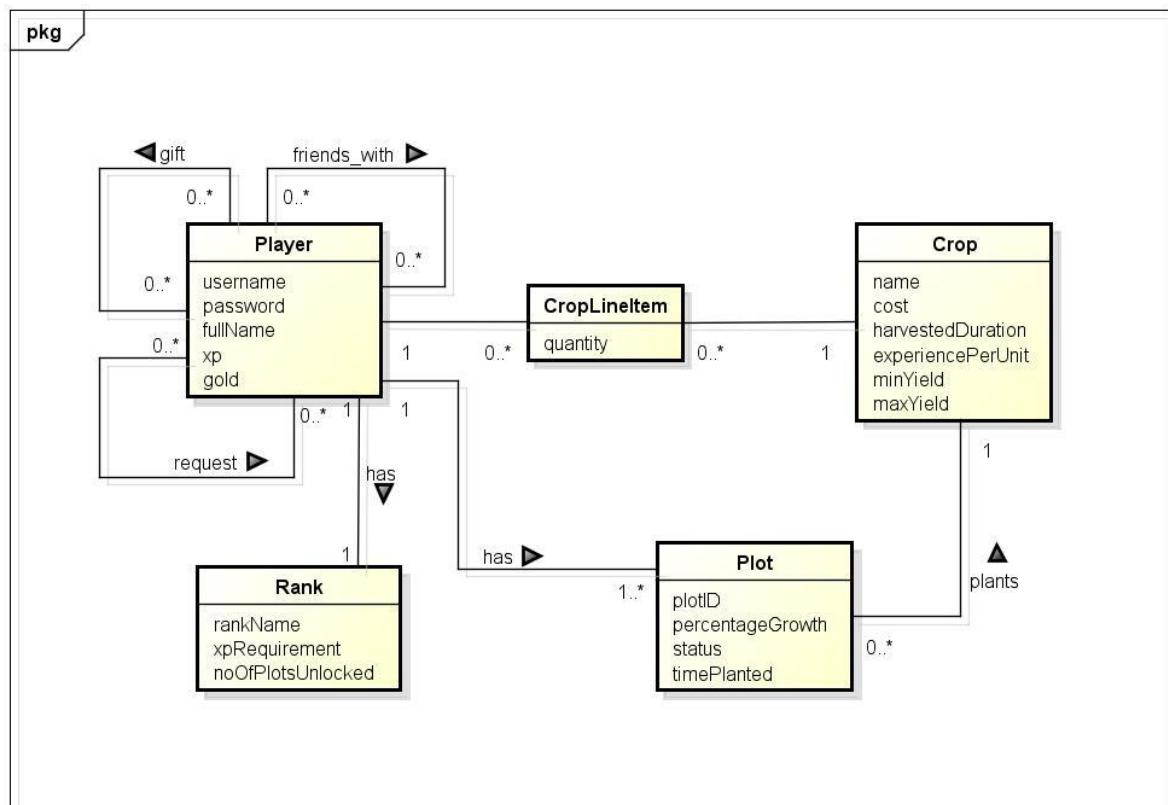
6.	Sequence Diagram	35
6.1.	Register.....	35
6.2.	Login	36
6.3.	Manage Friend	37
6.3.1.	Unfriend Friend	37
6.3.2.	Request Friend	38
6.3.3.	Accept Friend.....	39
6.3.4.	Reject Friend.....	40
6.4.	Manage Farm	41
6.4.1.	Plant Crop	41
6.4.2.	Clear Crop	42
6.4.3.	Harvest Crop.....	43
6.5.	Manage Inventory	44
6.5.1.	Buy Crop.....	44
6.5.2.	Gift Crop.....	45
7.	Design Considerations	46
8.	Challenges faced and lessons learnt	47
9.	Appendix.....	48

1. Use Case Diagram

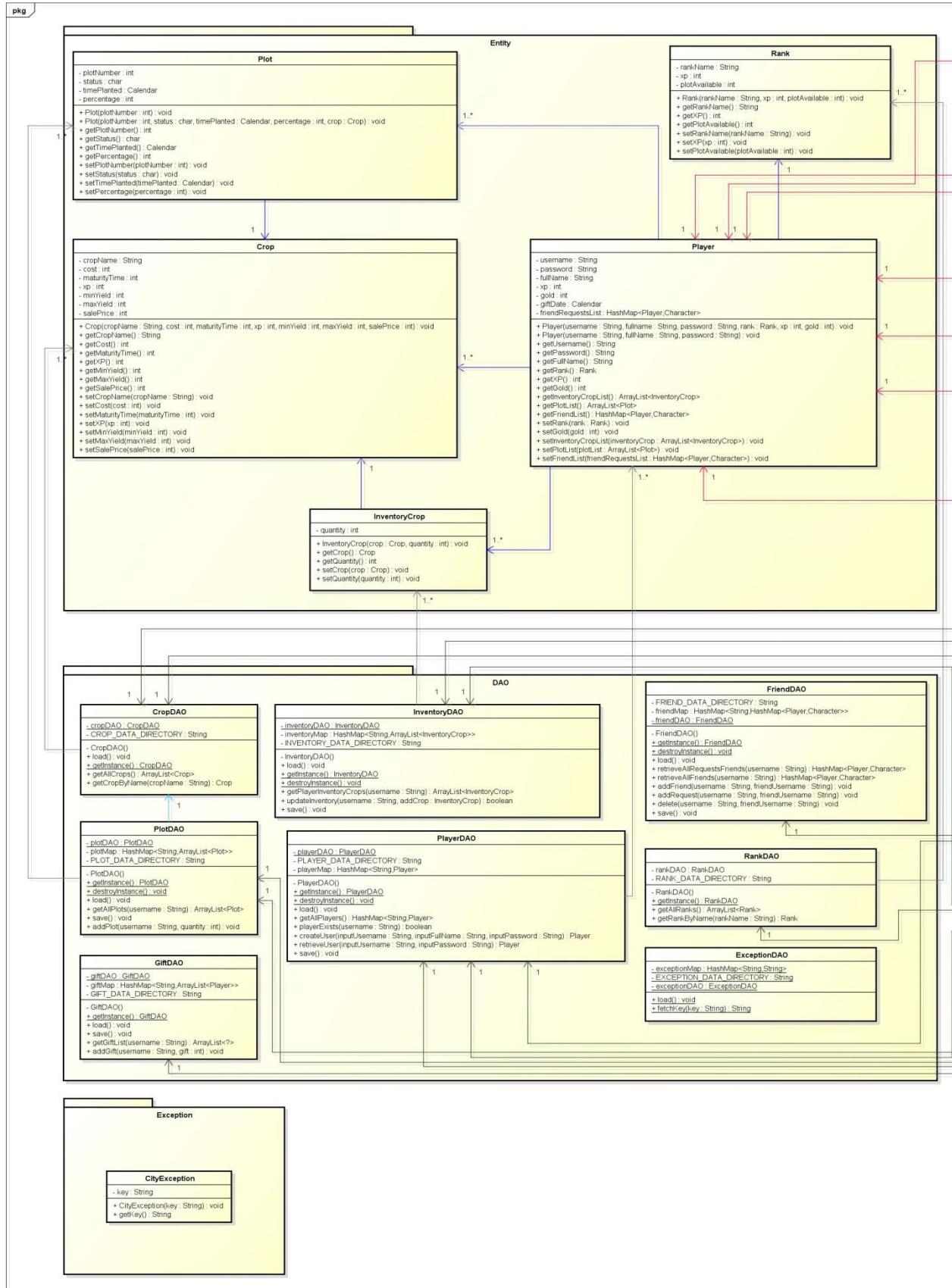


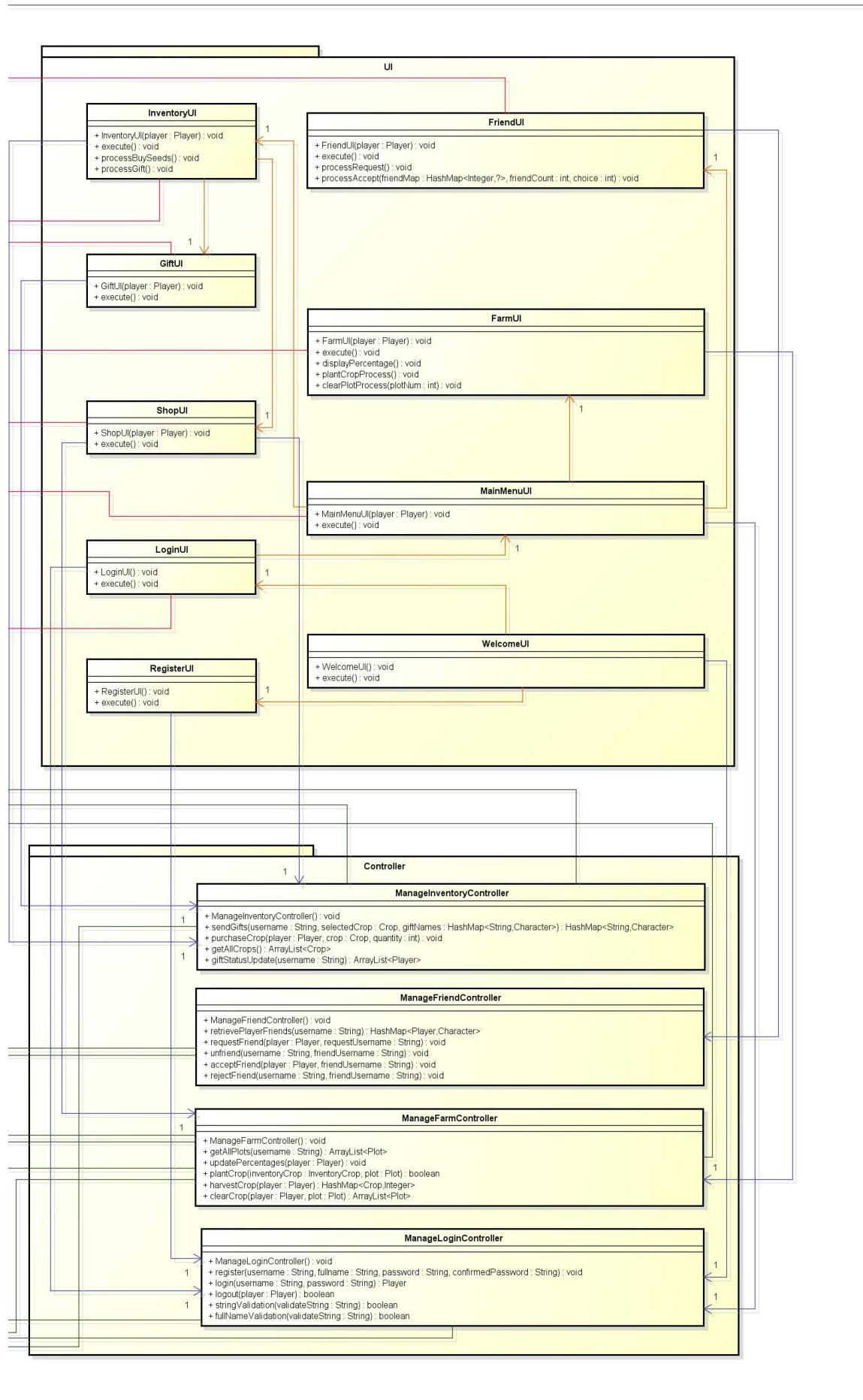
Use Case	Description
Register Account	This use case allows player who does not have an account with Farm City to register in order to play the game.
Login/Logout	This use case allows player to login into Farm City and log out from the game.
Manage Friends	This use case allows player to view friend, accept friend, reject friend, unfriend and send friend requests.
Manage Farm	This use case allows player to view farm, plant crops, harvest crops and clear wilted plots.
Manage Inventory	This use case allows player to view inventory, purchase bag of seeds and send gifts to their friends.

2. Domain Diagram



3. Class Diagram





4. Use Case Scenarios

4.1. Use Case – REGISTER ACCOUNT

Actor

Player

Precondition(s)

Player does not have an account and selected the registration option on the welcome page.

Main Flow of events

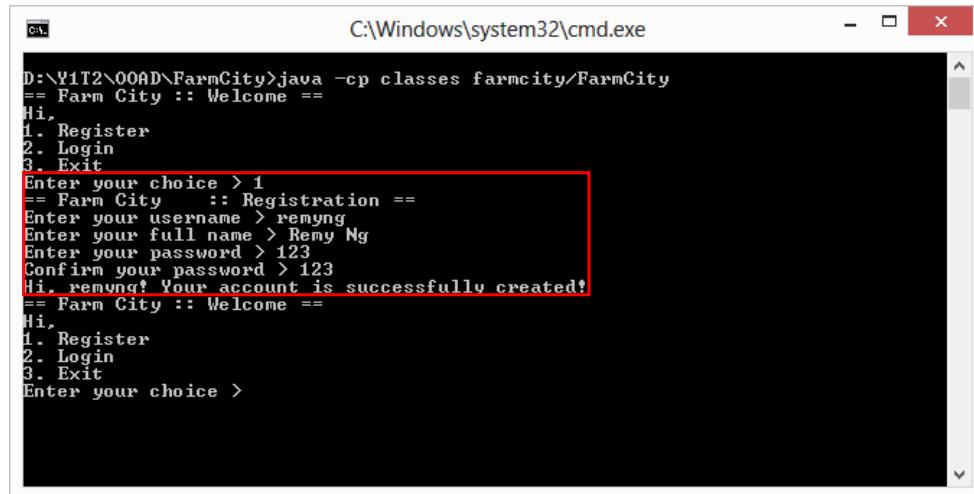
1. The use case begins with the player selects the register function on the welcome page
2. System prompts the player to enter his/her username, full name, password and confirm password
3. Player will give his/her inputs accordingly
4. System will check if username has been registered and validates it only contains alphanumeric characters. System will also validate and ensure that the input in the password and confirm password fields are the same. Upon successful registration, a new user account will be created with 0xp, 1000 gold coins and novice by rank.
5. System will re-display the welcome page.

Alternate flow of events

- 4a. Username has been registered
 - 1) System will prompt player to re-enter username
 - 2) Player enters a different username
 - 3) System will verify if username is taken. If username is taken, system will repeat Step 1 and 2 of the alternate flow of events until the player enters an unused username
 - 4) This will return to the main flow of events
- 4b. Password and confirm password is different
 - 1) System will prompt player to re-enter password and confirm password
 - 2) Player enters password and confirm password
 - 3) System will validate to make sure the input in both fields are similar
 - 4) This will return to the main flow of events

4.1.1. Screenshots

Registered Successfully

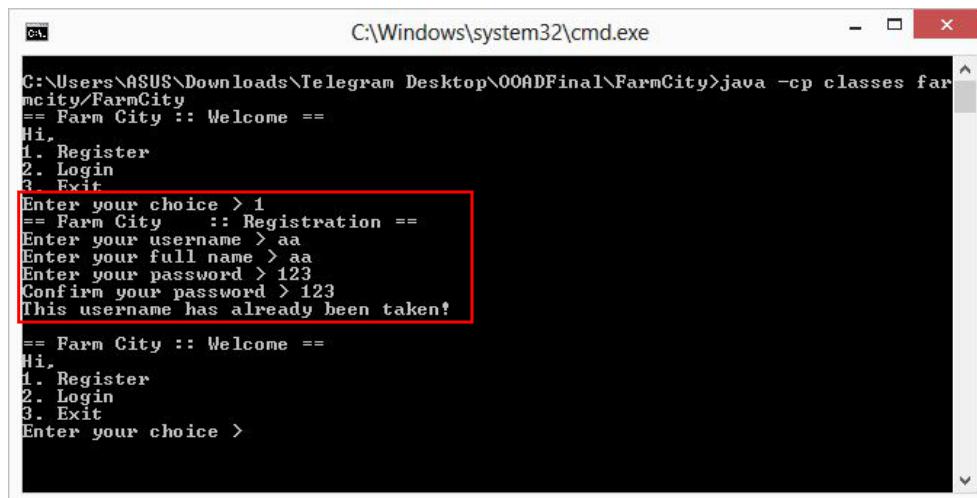


```
C:\Windows\system32\cmd.exe
D:\V1T2\OOAD\FarmCity>java -cp classes farmcity\FarmCity
== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice > 1
== Farm City :: Registration ==
Enter your username > remyng
Enter your full name > Remy Ng
Enter your password > 123
Confirm your password > 123
Hi, remyng! Your account is successfully created!
== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice >
```

The output shows the user entering choice 1 for registration. It prompts for a username ('remyng'), full name ('Remy Ng'), and password ('123'). Upon confirmation, it displays a success message: 'Hi, remyng! Your account is successfully created!'.

Unsuccessful Registration

- Username has been registered

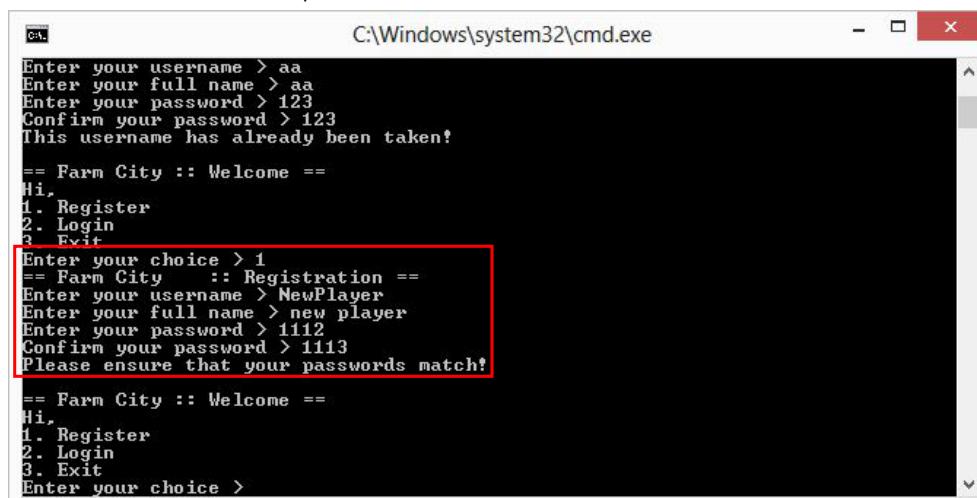


```
C:\Windows\system32\cmd.exe
C:\Users\ASUS\Downloads\Telegram Desktop\OOADFinal\FarmCity>java -cp classes farmcity\FarmCity
== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice > 1
== Farm City :: Registration ==
Enter your username > aa
Enter your full name > aa
Enter your password > 123
Confirm your password > 123
This username has already been taken!

== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice >
```

The output shows the user attempting to register with the username 'aa', which is already taken, resulting in an error message: 'This username has already been taken!'.

- Password and confirm password is different



```
C:\Windows\system32\cmd.exe
Enter your username > aa
Enter your full name > aa
Enter your password > 123
Confirm your password > 123
This username has already been taken!

== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice > 1
== Farm City :: Registration ==
Enter your username > NewPlayer
Enter your full name > new player
Enter your password > 1112
Confirm your password > 1113
Please ensure that your passwords match!

== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice >
```

The output shows the user attempting to register with a password ('1112') and confirmation ('1113') that do not match, resulting in an error message: 'Please ensure that your passwords match!'.

4.2. Use Case - LOGIN / LOGOUT

4.2.1. Use Case Scenario: Login

Actor

Player

Precondition(s)

Player must have a Farm City account and have not logged in.

Main Flow of Events

1. This use case begins when the player selects on the login option on the welcome page.
2. The system will prompt the player to enter his/her username and password
3. Player enters username and password
4. System verifies if the username is valid and the password matches the username. If the authentication succeed, the system will display the Game Menu of Farm City

Alternate Flow of Events

- 4a. If username/password is not valid
- 1) System will display an error message showing “incorrect username and password”
 - 2) This will return to Step 2 of the main flow diagram

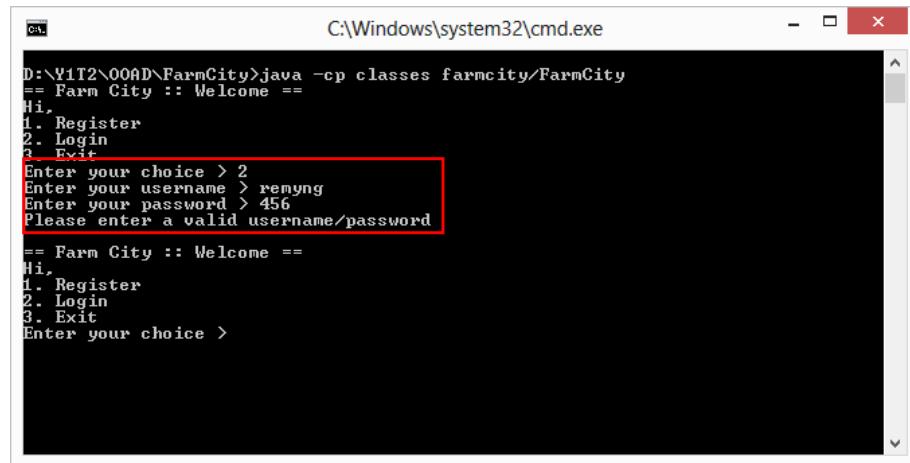
4.2.1.1. Screenshots

Successfully login

```
C:\Windows\system32\cmd.exe
2. Login
3. Exit
Enter your choice > 1
== Farm City :: Registration ==
Enter your username > remyng
Enter your full name > Remy Ng
Enter your password > 123
Confirm your password > 123
Hi, remyng! Your account is successfully created!
== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice > 2
Enter your username > remyng
Enter your password > 123
== Farm City :: Main Menu ==
Welcome, Remy Ng
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice >
```

Unsuccessful Login

- If username/password is not valid



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command entered is 'java -cp classes farmcity/FarmCity'. The output shows a welcome message and a menu with options 1. Register, 2. Login, and 3. Exit. The user enters '2' for Login. Then, they enter 'remyng' for the username and '456' for the password. A red box highlights the error message 'Please enter a valid username/password'.

```
D:\YIT2\OOAD\FarmCity>java -cp classes farmcity/FarmCity
== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice > 2
Enter your username > remyng
Enter your password > 456
Please enter a valid username/password

== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice >
```

4.2.2. Use Case Scenario: Logout

Actor

Player

Precondition(s)

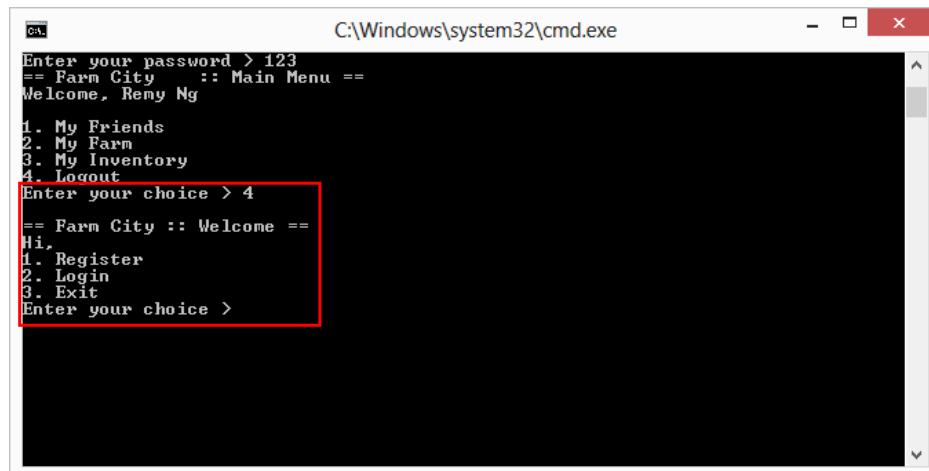
Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the logout option on the Game Menu of Farm City
2. The system will direct player to the welcome page

4.2.2.1. Screenshots

Successfully logout



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command entered is 'java -cp classes farmcity/FarmCity'. The output shows a welcome message for 'Remy Ng' and a menu with options 1. My Friends, 2. My Farm, 3. My Inventory, and 4. Logout. The user enters '4' for Logout. A red box highlights the message '== Farm City :: Welcome ==' which appears again after the logout.

```
Enter your password > 123
== Farm City :: Main Menu ==
Welcome, Remy Ng
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 4

== Farm City :: Welcome ==
Hi,
1. Register
2. Login
3. Exit
Enter your choice >
```

4.3. Use Case – MANAGE FRIENDS

4.3.1. Use Case Scenario: Unfriend Friend

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the unfriend option ‘U’ with the <ID> in the “My Friends” Menu (e.g. U1)
2. The system will remove the user from player’s friend list and player from user’s friend list
3. The system will return a success message to player and display the “My Friends” menu

Alternate Flow of Events

2a. If <ID> is not valid

- 1) System will display an error message showing “You have entered an invalid option.”
- 2) This will end the use case and displays the “My Friends” menu

4.3.1.1. Screenshots

Successfully unfriend a friend

```
C:\Windows\system32\cmd.exe
Rank : Novice    Gold : 1000
My seeds:
1.4 Bags of Papaya
[M]ain : [B]uy : [G]ift : Select choice > M
Please enter a valid input!
== Farm City      :: Main Menu ==
Welcome, kk

1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, kk

My Friends: My Friends:
1.aa
My Requests:

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > U1
```

Failed to unfriend a friend

- If <ID> is not valid

```

C:\Windows\system32\cmd.exe
2.kk
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > A2
kk is now your friend.

== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kaka
2.kk
My Requests:

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > U3
Please ensure that you entered a valid input!

== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kaka
2.kk
My Requests:

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >

```

4.3.2. Use Case Scenario: Request Friend

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the request option [Q] in the “My Friends” Menu
2. The system will prompt the player to enter the username of the player he/she wants to add
3. The player will enter the username of the friend
4. The system will search for the request friend’s username and send a request to the player with that particular username
5. The system will return a success message to player and display the “My Friends” menu

Alternate Flow of Events

- 4a. If username is not valid or player enters his/her own username
 - 1) System will display an error message showing “Please ensure that you have entered a valid user.”
 - 2) This will end the use case and displays the “My Friends” menu
- 4b. If username is already a friend
 - 1) System will display an error message showing “You are already friends with <username>.”
 - 2) This will end the use case and displays the “My Friends” menu

- 4c. If player has already sent a request to that particular username
- 1) System will display an error message showing “You already have a pending friend request for <username>.”
 - 2) This will end the use case and displays the “My Friends” menu
- 4d. If the player has already received a request from that particular username
- 1) System will display an error message showing “You already have a pending friend request for <username>.”
 - 2) This will end the use case and displays the “My Friends” menu

4.3.2.1. Screenshots

Successfully sent a friend request

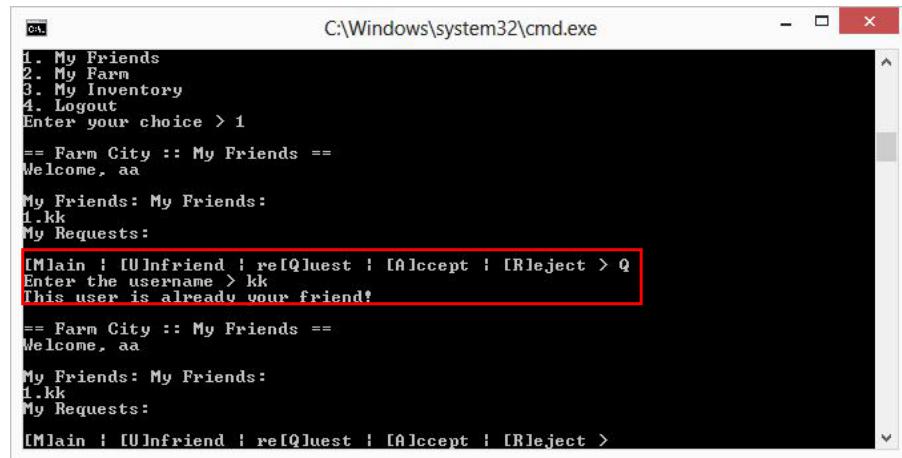
```
C:\Windows\system32\cmd.exe
Welcome, Remy Ng
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, Remy Ng
My Friends: My Friends:
My Requests:
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
Enter the username > aa
You have sent aa a request
== Farm City :: My Friends ==
Welcome, Remy Ng
My Friends: My Friends:
My Requests:
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

Failed to send a friend request

- If username is not valid

```
C:\Windows\system32\cmd.exe
My Friends:
My Requests :
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
Enter the username >
chloe
You have sent chloe a request
== Farm City :: My Friends ==
Welcome, Remy Ng
My Friends:
My Requests :
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
barackObama
Please ensure that you entered a valid input!
== Farm City :: My Friends ==
Welcome, Remy Ng
My Friends:
My Requests :
[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

- If username is already a friend



```
C:\Windows\system32\cmd.exe
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kk
My Requests:

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
Enter the username > kk
This user is already your friend!

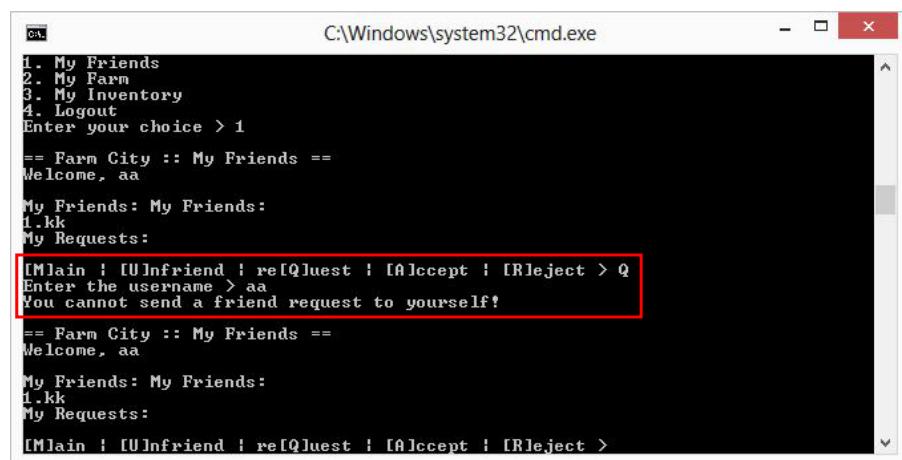
== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kk
My Requests:

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

The screenshot shows a command-line interface for a game named 'Farm City'. The user has chosen to view their friends. They enter 'kk' as a friend's username. A red box highlights the response: 'This user is already your friend!', indicating that the user cannot send a friend request to themselves.

- If player has already send a request to that particular username



```
C:\Windows\system32\cmd.exe
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kk
My Requests:

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
Enter the username > aa
You cannot send a friend request to yourself!

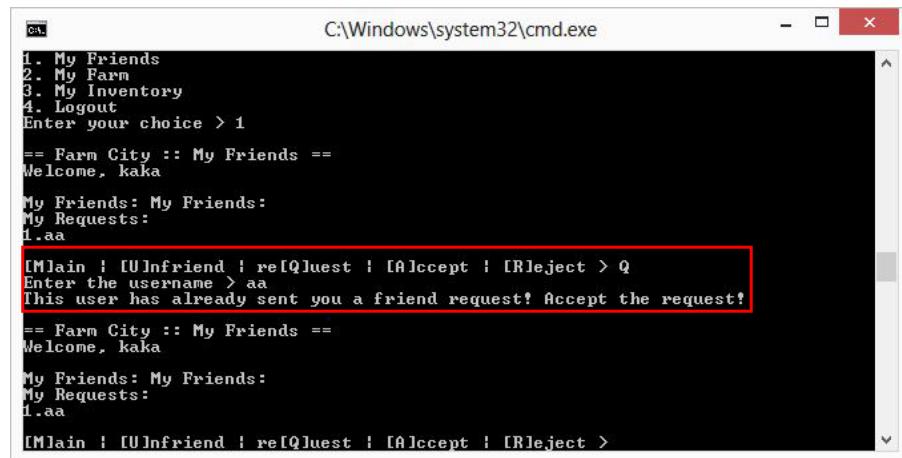
== Farm City :: My Friends ==
Welcome, aa

My Friends: My Friends:
1.kk
My Requests:

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

The screenshot shows a command-line interface for a game named 'Farm City'. The user has chosen to view their friends. They enter 'aa' as a friend's username. A red box highlights the response: 'You cannot send a friend request to yourself!', indicating that the user cannot send a friend request to themselves.

- If the player has already received a request from that particular username



```
C:\Windows\system32\cmd.exe
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, kaka

My Friends: My Friends:
My Requests:
1.aa

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > Q
Enter the username > aa
This user has already sent you a friend request! Accept the request!

== Farm City :: My Friends ==
Welcome, kaka

My Friends: My Friends:
My Requests:
1.aa

[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

The screenshot shows a command-line interface for a game named 'Farm City'. The user has chosen to view their friends. They enter 'aa' as a friend's username. A red box highlights the response: 'This user has already sent you a friend request! Accept the request!', indicating that the user has already received a friend request from themselves.

4.3.3. Use Case Scenario: Accept Friend

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the accept option with the <ID> in the “My Friends” Menu (e.g. A2)
2. The system will accept the chosen friend request in the request list
3. The system will return a success message and display the “My Friends” menu

Alternate Flow of Events

- 2a. If <ID> is not valid
 - 1) System will display an error message showing “You have entered an invalid option.”
 - 2) This will end the user case and displays the “My Friends” menu

4.3.3.1. Screenshots

Successfully accepted a friend request

```
C:\Windows\system32\cmd.exe
== Farm City    :: Main Menu ==
Welcome, 123
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, 123

My Friends:
My Requests :
1.remyng
[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > A1
remyng is now your friend.
== Farm City .. My Friends ==
Welcome, 123

My Friends:
1.remyng
My Requests :
[Main : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

Fail to accept a friend request

- If <ID> is not valid

```

C:\Windows\system32\cmd.exe
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, Remy Ng

My Friends:
1.chloe
2.Uu
My Requests :
3.superman

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
Please ensure that you entered a valid input!
== Farm City :: My Friends ==
Welcome, Remy Ng

My Friends:
1.chloe
2.Uu
My Requests :
3.superman

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >

```

4.3.4. Use Case Scenario: Reject Friend

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the reject option with the <ID> in the “My Friends” Menu
2. The system will reject the friend request in the request list
3. The system will return a success message and displays the “My Friends” menu

Alternate Flow of Events

- 2a. If <ID> is not valid

- 1) System will display an error message showing “You have entered an invalid option.”
- 2) This will end the use case and displays the “My Friends” menu

4.3.4.1. Screenshots

Successfully rejected a friend request

```
Welcome, SUPAMAN
1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, SUPAMAN

My Friends:
My Requests :
1.remyng
2.superwoman

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > R2
You have rejected the request of superwoman
== Farm City :: My Friends ==
Welcome, SUPAMAN

My Friends:
My Requests :
1.remyng

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

Failed to reject a friend request

- If <ID> is not valid

```
Welcome, SUPAMAN
== Farm City :: Main Menu ==
Welcome, SUPAMAN

1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1
== Farm City :: My Friends ==
Welcome, SUPAMAN

My Friends:
My Requests :
1.remyng

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject > R2
Invalid option!
== Farm City :: My Friends ==
Welcome, SUPAMAN

My Friends:
My Requests :
1.remyng

[M]ain : [U]nfriend : re[Q]uest : [A]ccept : [R]eject >
```

4.4. Use Case – MANAGE FARM

4.4.1. Use Case Scenario: Plant Crop

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the plant option [P] with the selected plot <ID> in the “My Farm” Menu (e.g P1)
2. The system will return the list of crops that the player has in their inventory
3. The system will prompt the player to enter the <ID> of the crop that they wish to plant
4. The player will select the crop option
5. The system will remove 1 bag of seeds of the selected crop from player’s inventory
6. The system will display a message informing the player that the crop has been successfully planted and display “My Farm” Menu.

Alternate Flow of Events

2a. Player enters an invalid plot number

- 1) System will return an error message showing “Invalid plot number.”
- 2) This will end the use case and displays the “My Farm” menu

2b. Player enters an option with a planted/wilted crop

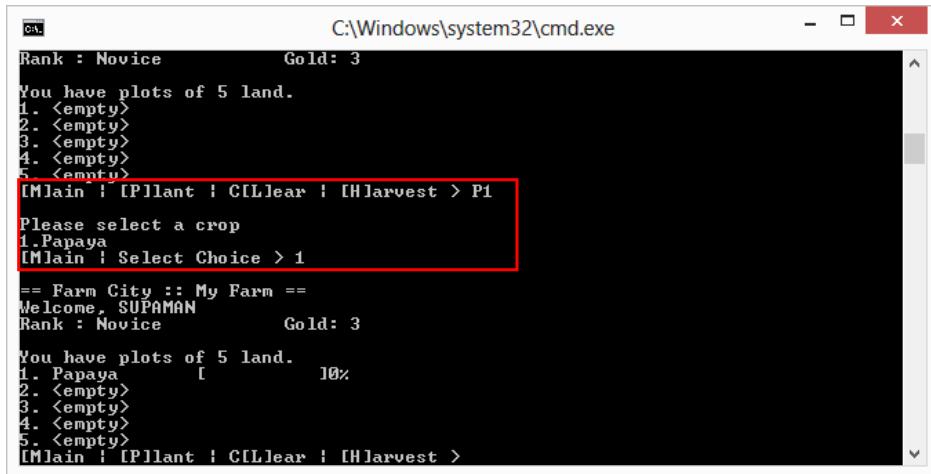
- 1) System will return an error message showing “Plot is unavailable.”
- 2) This will end the use case and displays the “My Farm” menu

5a. Player enters an invalid crop number

- 1) System will return an error message showing “Invalid crop number”
- 2) This will return to Step 3 of the main flow of events

4.4.1.1. Screenshots

Successfully planted a crop



C:\Windows\system32\cmd.exe

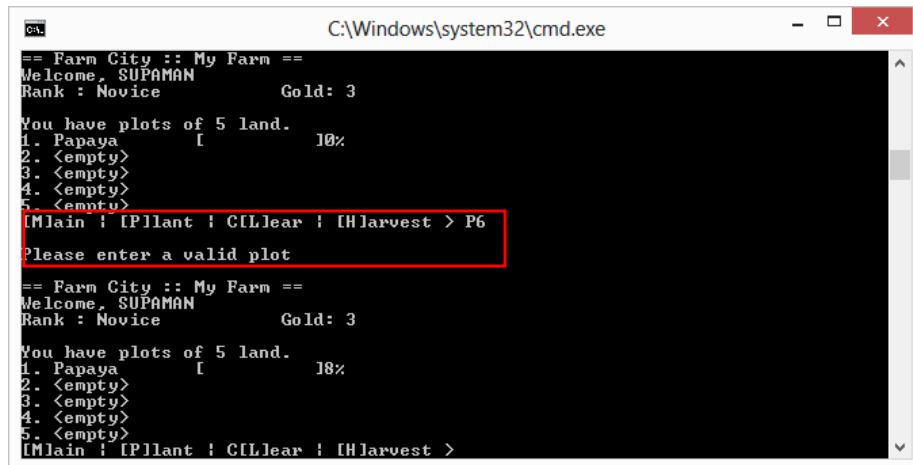
```

Rank : Novice          Gold: 3
You have plots of 5 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > P1
Please select a crop
1.Papaya
[M]ain : Select Choice > 1
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice          Gold: 3
You have plots of 5 land.
1. Papaya      [          10%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >

```

Failed to plant a crop

- Player enters an invalid plot number



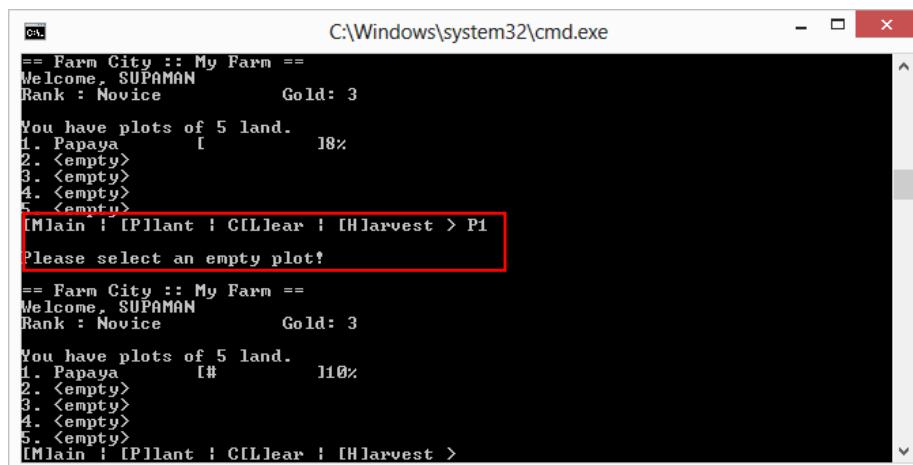
C:\Windows\system32\cmd.exe

```

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice          Gold: 3
You have plots of 5 land.
1. Papaya      [          10%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > P6
Please enter a valid plot
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice          Gold: 3
You have plots of 5 land.
1. Papaya      [          10%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >

```

- Player enters an option with a planted/wilted crop



C:\Windows\system32\cmd.exe

```

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice          Gold: 3
You have plots of 5 land.
1. Papaya      [          10%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > P1
Please select an empty plot!
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice          Gold: 3
You have plots of 5 land.
1. Papaya      [#          10%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >

```

- Player enters an invalid crop number

```

C:\Windows\system32\cmd.exe
C:\>
3. <empty>
4. <empty>
5. <empty>
[MLain : LPPlant : CILClear : IHHarvest > P1
Please select an empty plot!
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice           Gold: 3
You have plots of 5 land.
1. Papaya      [# 110%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[MLain : LPPlant : CILClear : IHHarvest > P2
Please select a crop
1.Papaya
[MLain : Select Choice > 2
Please select a crop
1.Papaya
[MLain : Select Choice >

```

4.4.2. Use Case Scenario: Clear Crop

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the clear option [L] with the selected plot <ID>in the “My Farm” Menu (e.g. L1)
2. The system will search for the specified plot
3. The system will deduct 5 gold from player’s account and clear the plot
4. The system will return a success message to the player and display “My Farm” Menu.

Alternate Flow of Events

2a. Player enters invalid plot number

- 1) System will return a message showing “Invalid plot number”
- 2) System will exit the use case and displays the “My Farm” menu

2b. Player selects a plot that does not have a wilted crop

- 1) System will return a message showing “No wilted crops”
- 2) System will exit the use case and displays the “My Farm” menu

3a. If player has insufficient gold

- 1) System will return a message showing “Insufficient gold”
- 2) System will exit the use case and displays the “My Farm” menu

4.4.2.1. Screenshots

Successfully cleared crop

```
C:\Windows\system32\cmd.exe
Enter your choice > 2
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice Gold: 735
You have plots of 5 land.
1. Papaya [ Wilted ]
2. Papaya [ Wilted ]
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > L1
Plot has been cleared!

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice Gold: 730
You have plots of 5 land.
1. <empty>
2. Papaya [ Wilted ]
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >
```

Failed to clear crop

- Player enters invalid plot number

```
C:\Windows\system32\cmd.exe
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice Gold: 730
You have plots of 5 land.
1. <empty>
2. Papaya [ Wilted ]
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > L6
Please enter a valid plot number

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice Gold: 730
You have plots of 5 land.
1. <empty>
2. Papaya [ Wilted ]
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >
```

- Player selects a plot that does not have a wilted crop

```
C:\Windows\system32\cmd.exe
Enter your choice > 2
== Farm City :: My Farm ==
Welcome, Chloe Chan
Rank : Novice Gold: 1000
You have plots of 5 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > L1
No wilted crops to clear!

== Farm City :: My Farm ==
Welcome, Chloe Chan
Rank : Novice Gold: 1000
You have plots of 5 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >
```

- If player has insufficient gold

```

Enter your choice > 2
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice           Gold: 3
You have plots of 5 land.
1. <empty>
2. Papaya      [ Wilted   ]
3. Papaya      [ Wilted   ]
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > L2
All wilted plot(s) have been cleared free of charge.

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice           Gold: 3
You have plots of 5 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >

```

4.4.3. Use Case Scenario: Harvest Crop

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the harvest option [H] in the “My Farm” Menu.
2. The system will search through the list of plots for crops that are ready for harvesting (100% fully grown) on any plot of land
3. The system will calculate the total gold obtained from the number of units produced and add the gold + XP into the player’s account
4. The system will return a success message to the player and display “My Farm” Menu.

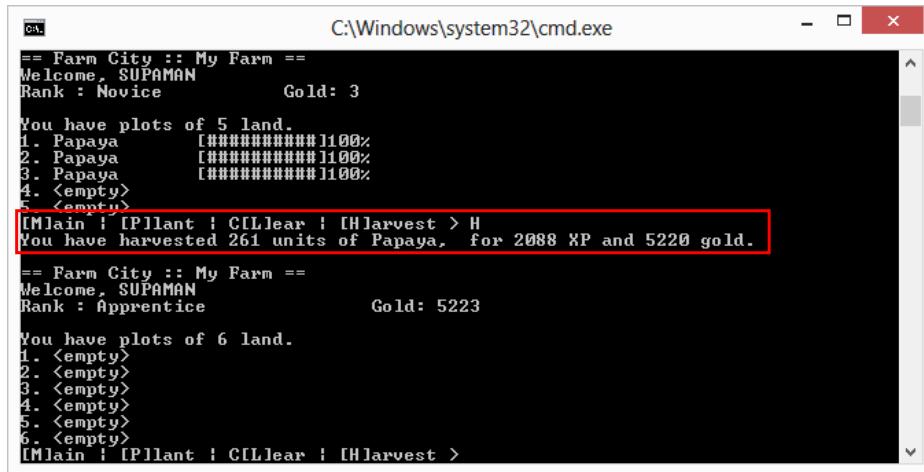
Alternate Flow of Events

2a. Player enters harvest option but there is no crop to be harvest

- 1) System will return a message showing “No crops to harvest”
- 2) System will exit the use case and display “My Farm” Menu.

4.4.3.1. Screenshots

Successfully harvest crops



```
C:\Windows\system32\cmd.exe
== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Novice Gold: 3

You have plots of 5 land.
1. Papaya [#####]100%
2. Papaya [#####]100%
3. Papaya [#####]100%
4. <empty>
5. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > H
You have harvested 261 units of Papaya, for 2088 XP and 5220 gold.

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Apprentice Gold: 5223

You have plots of 6 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
6. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >
```

Failed to harvest crops

- Player enters harvest option but there is no crop to be harvest



```
C:\Windows\system32\cmd.exe
Welcome, SUPAMAN
Rank : Apprentice Gold: 5223

You have plots of 6 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
6. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest > H
Nothing to harvest!

== Farm City :: My Farm ==
Welcome, SUPAMAN
Rank : Apprentice Gold: 5223

You have plots of 6 land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
6. <empty>
[M]ain : [P]lant : [C]lear : [H]arvest >
```

4.5. Use Case – MANAGE INVENTORY

4.5.1. Buy Crop

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the buy option [B] in the “My Inventory” Menu.
2. The system will list down all the available seeds that are available for sale and their information
3. The system will prompt the player to enter which seed they wish to purchase
4. The player will enter the seed <ID> from the list of available seeds
5. The system will prompt the player to enter the quantity of bag(s) of seeds
6. The player will enter the quantity they wish to purchase
7. The system will calculate the total cost of the bag(s) of seeds, deduct the total cost from the player’s account and add the bag(s) of seeds into the player’s inventory
8. The system will return a success message to the player and display “My Inventory” Menu.

Alternate Flow of Events

5a. Player enters an invalid seed option

- 1) System will return a message showing “Invalid option”
- 2) This will return to Step 3 of the main flow diagram.

5b. Player enter ‘M’ after he/she chooses to buy

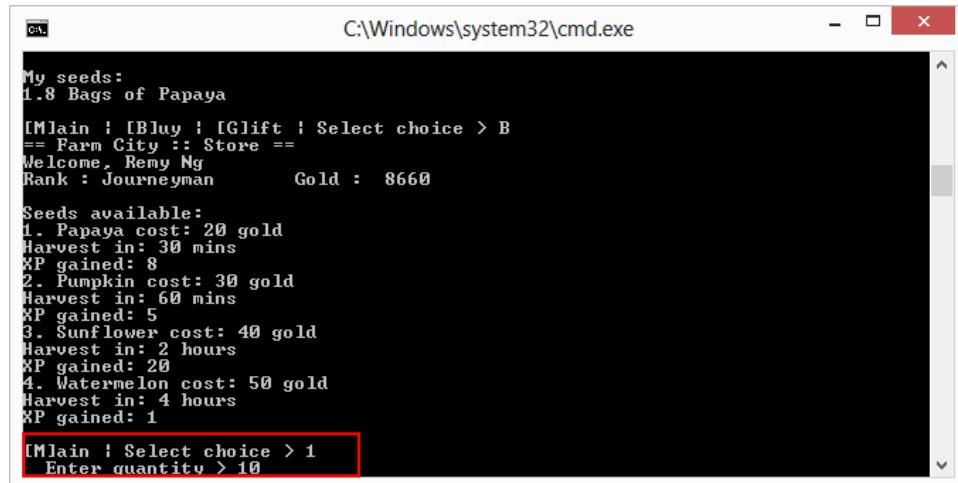
- 1) This use case will end and display “My Inventory” Menu

7a. If the player has insufficient gold

- 1) The system will return a message to showing “Insufficient gold”
- 2) This use case will end and display “My Inventory” Menu

4.5.1.1. Screenshots

Successfully buy bag(s) of seeds

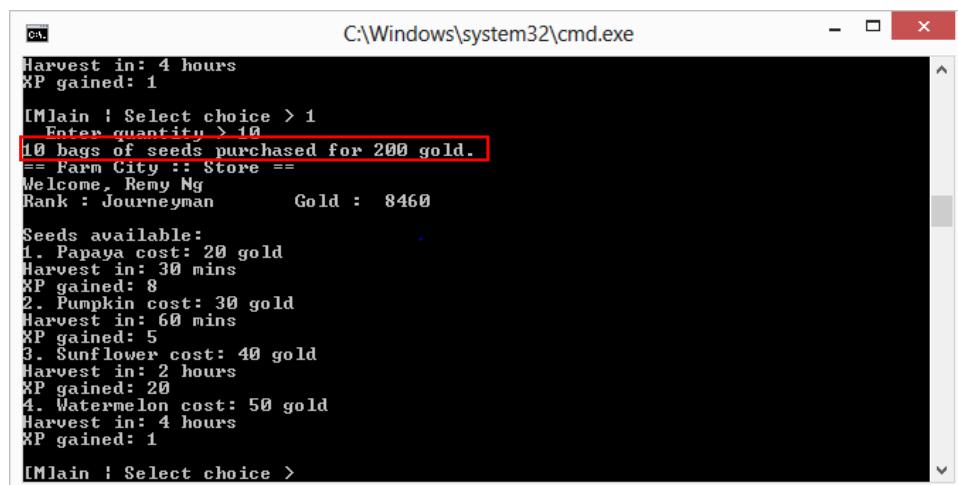


```
C:\Windows\system32\cmd.exe
My seeds:
1.8 Bags of Papaya

[MLain : [B]Buy : [G]ift : Select choice > B
== Farm City :: Store ==
Welcome, Remy Ng
Rank : Journeyman Gold : 8660

Seeds available:
1. Papaya cost: 20 gold
Harvest in: 30 mins
XP gained: 8
2. Pumpkin cost: 30 gold
Harvest in: 60 mins
XP gained: 5
3. Sunflower cost: 40 gold
Harvest in: 2 hours
XP gained: 20
4. Watermelon cost: 50 gold
Harvest in: 4 hours
XP gained: 1

[MLain : Select choice > 1
Enter quantity > 10
```



```
C:\Windows\system32\cmd.exe
Harvest in: 4 hours
XP gained: 1

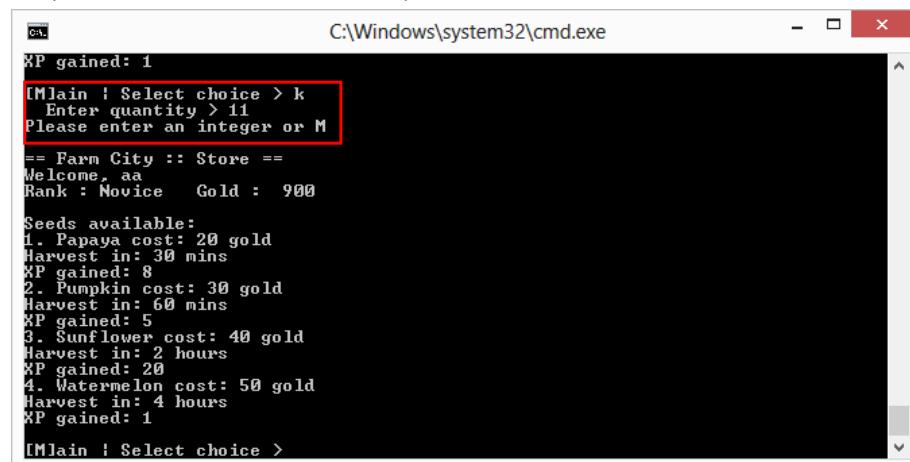
[MLain : Select choice > 1
Enter quantity > 10
10 bags of seeds purchased for 200 gold.
== Farm City :: Store ==
Welcome, Remy Ng
Rank : Journeyman Gold : 8460

Seeds available:
1. Papaya cost: 20 gold
Harvest in: 30 mins
XP gained: 8
2. Pumpkin cost: 30 gold
Harvest in: 60 mins
XP gained: 5
3. Sunflower cost: 40 gold
Harvest in: 2 hours
XP gained: 20
4. Watermelon cost: 50 gold
Harvest in: 4 hours
XP gained: 1

[MLain : Select choice >
```

Failed to buy bag(s) of seeds

- Player enters an invalid seed option



```
C:\Windows\system32\cmd.exe
XP gained: 1

[MLain : Select choice > k
Enter quantity > 11
Please enter an integer or M

== Farm City :: Store ==
Welcome, aa
Rank : Novice Gold : 900

Seeds available:
1. Papaya cost: 20 gold
Harvest in: 30 mins
XP gained: 8
2. Pumpkin cost: 30 gold
Harvest in: 60 mins
XP gained: 5
3. Sunflower cost: 40 gold
Harvest in: 2 hours
XP gained: 20
4. Watermelon cost: 50 gold
Harvest in: 4 hours
XP gained: 1

[MLain : Select choice >
```

- Player enter 'M' after he/she chooses to buy

```

Seeds available:
1. Papaya cost: 20 gold
Harvest in: 30 mins
XP gained: 8
2. Pumpkin cost: 30 gold
Harvest in: 60 mins
XP gained: 5
3. Sunflower cost: 40 gold
Harvest in: 2 hours
XP gained: 20
4. Watermelon cost: 50 gold
Harvest in: 4 hours
XP gained: 1
[M]ain : Select choice > M

== Farm City :: My Inventory ==
Welcome, aa
Rank : Novice Gold : 900
My seeds:
1.4 Bags of Papaya
[M]ain : [B]Buy : [G]ift : Select choice >

```

- If the player has insufficient gold

```

1. Papaya cost: 20 gold
Harvest in: 30 mins
XP gained: 8
2. Pumpkin cost: 30 gold
Harvest in: 60 mins
XP gained: 5
3. Sunflower cost: 40 gold
Harvest in: 2 hours
XP gained: 20
4. Watermelon cost: 50 gold
Harvest in: 4 hours
XP gained: 1
[M]ain : Select choice > 1
Enter quantity > 500
Insufficient gold.

== Farm City :: My Inventory ==
Welcome, kk
Rank : Novice Gold : 1000
My seeds:
1.4 Bags of Papaya
[M]ain : [B]Buy : [G]ift : Select choice >

```

4.5.2. Gift Crop

Actor

Player

Precondition(s)

Player must have a Farm City account and have logged in.

Main Flow of Events

1. This use case begins when the player selects the send a gift option in the "My Inventory" Menu.
2. The system will validate if player has already sent 5 gifts on that day
3. The system will list down the available gifts from the game
4. The system prompts player to enter the bag of seeds that he/she wishes to send their friend(s)
5. The player will enter the seed option <ID>
6. The system prompts player to enter his/her friends' username(s)
7. The player enters the username(s)
8. The system will verify the username(s)

9. The system will check that player does not exceed the maximum number of gifts if he/she enters more than 2 usernames
10. The system will verify that player has not sent any gift to that user on that day
11. The system will add one bag of seeds into each friend's inventory and deduct the corresponding bag of seeds from the player's inventory
12. The system will update the number of gift(s) that player has been successfully sent on that day
13. The system will return a success message and display "My Inventory" Menu.

Alternate Flow of Events

2a. If player has sent more than 5 gifts

- 1) System will return a message showing "Gift(s) were not sent to <username>"
- 2) This will end the user case and displays the "My Inventory" menu

6a. Player enters an invalid seed option

- 1) System will return a message showing "Gift(s) were not sent to <username>"
- 2) This will end the user case and displays the "My Inventory" menu

6b. Players enters an option that he/she doesn't have in his/her inventory

- 1) System will return a message showing "Gift(s) were not sent to <username>"
- 2) This will end the user case and displays the "My Inventory" menu

8a. Player enters incorrect username or his/her own username

- 1) The system will return a message showing "Invalid user"
- 2) This will end the user case and displays the "My Inventory" menu

9a. If the number of users entered will exceed the maximum number of gifts per day

- 1) The system will add one bag of seeds to the 1st X user in the input (where X = 5 – number of gifts sent out that day) and deduct the corresponding bag of seeds from the player's inventory
- 2) The system return a message showing "You have sent gift to <username1>, <username2>. You are unable to send your gift to <username3>."
- 3) This will end the user case and displays the "My Inventory" menu

10a. Player has already sent one gift to the user that day

- 1) The system will return a message showing “You can only send one gift a day”
- 2) This will end the user case and displays the “My Inventory” menu

4.5.2.1. Screenshots

Successfully sent a gift to friend(s)

```

C:\Windows\system32\cmd.exe
XP gained: 1
[M]ain : Select choice > 1
Enter quantity > 500
Insufficient gold.

== Farm City :: My Inventory ==
Welcome, kk
Rank : Novice Gold : 1000

My seeds:
1.4 Bags of Papaya

[M]ain : [B]uy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, kk
Rank: Novice Gold:1000

Gifts Available:
1. 1 Bag of PapayaSeeds
2. 1 Bag of PumpkinSeeds
3. 1 Bag of SunflowerSeeds
4. 1 Bag of WatermelonSeeds
[M]ain : Select choice > 1
Send to>

```

```

C:\Windows\system32\cmd.exe
1.1001 Bags of Papaya

[M]ain : [B]uy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, aa
Rank: Novice Gold:900

Gifts Available:
1. 1 Bag of PapayaSeeds
2. 1 Bag of PumpkinSeeds
3. 1 Bag of SunflowerSeeds
4. 1 Bag of WatermelonSeeds
[M]ain : Select choice > 1
Send to> kaka
Gift(s) successfully sent to kaka

== Farm City :: My Inventory ==
Welcome, aa
Rank : Novice Gold : 900

My seeds:
1.1000 Bags of Papaya

[M]ain : [B]uy : [G]ift : Select choice >

```

Failed to send a gift to friend(s)

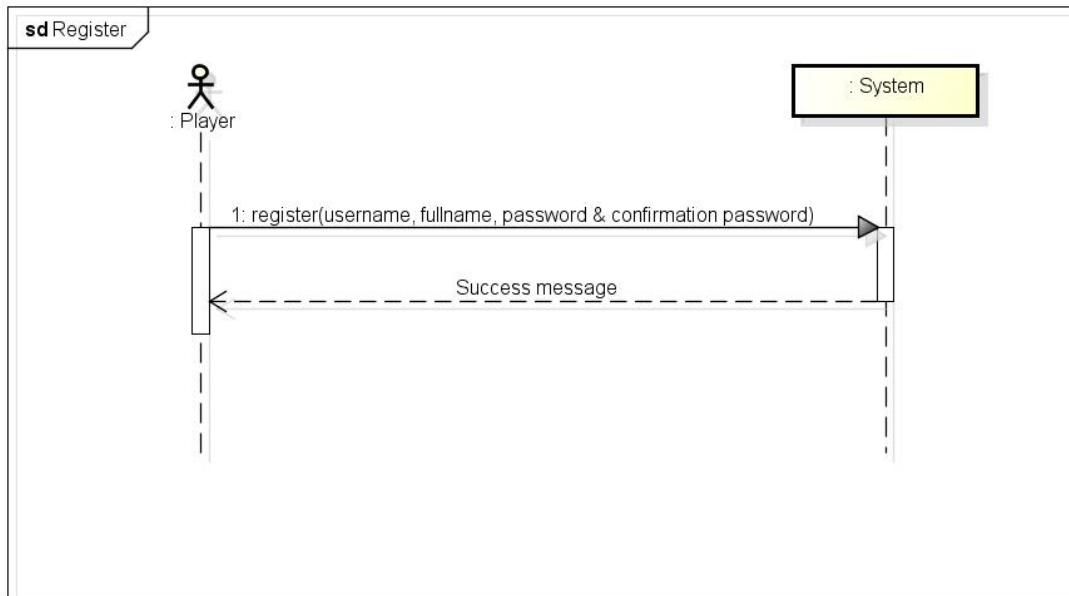
- Scenarios:
 - 1) If player has sent more than 5 gifts
 - 2) Player enters an invalid seed option
 - 3) Player enters an option that he/she doesn't have in his/her inventory
 - 4) Player enters incorrect username or his/her own username
 - 5) If the number of users entered will exceed the maximum number of gifts per day
 - 6) Player has already sent one gift to the user that day

```
C:\Windows\system32\cmd.exe
== Farm City :: Send a Gift ==
Welcome, kaka
Rank: Novice      Gold:725
Gifts Available:
1. 1 Bag of PapayaSeeds
2. 1 Bag of PumpkinSeeds
3. 1 Bag of SunflowerSeeds
4. 1 Bag of WatermelonSeeds
[Main : [Buy : [Gift : Select choice > 4
Send to> aa
Gift(s) were not sent to aa

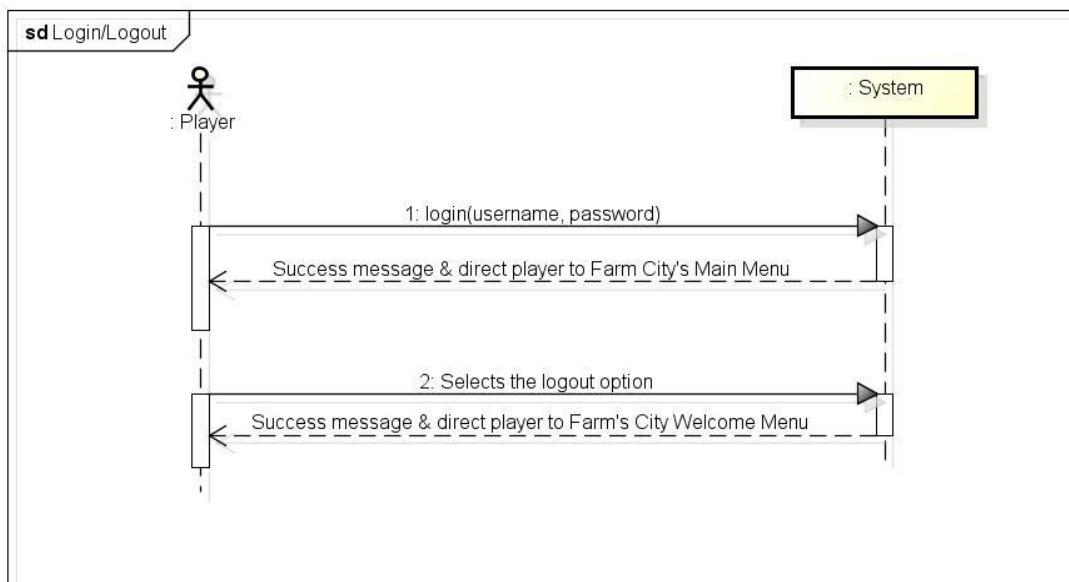
== Farm City :: My Inventory ==
Welcome, kaka
Rank : Novice      Gold :  725
My seeds:
1.1 Bags of Pumpkin
2.1 Bags of Watermelon
[Main : [Buy : [Gift : Select choice >
```

5. System Sequence Diagram

5.1. Register

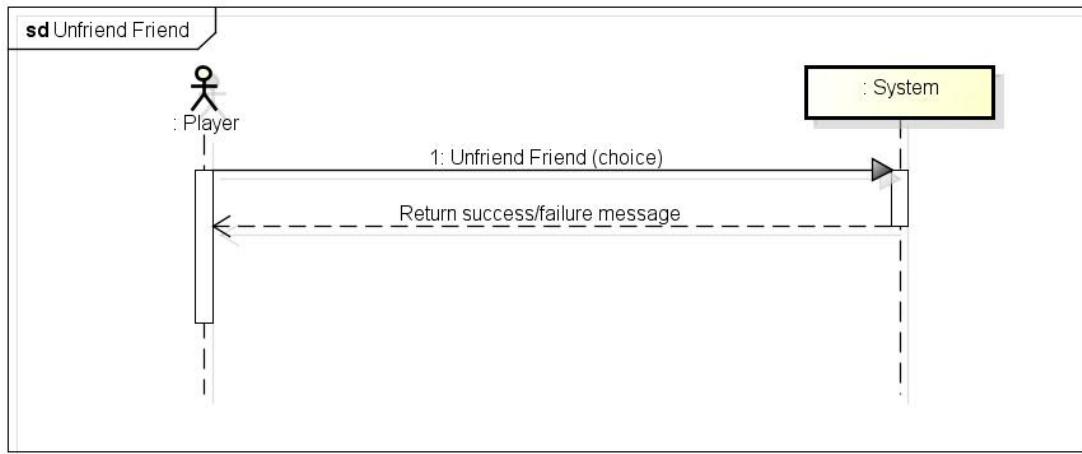


5.2. Login / Log out

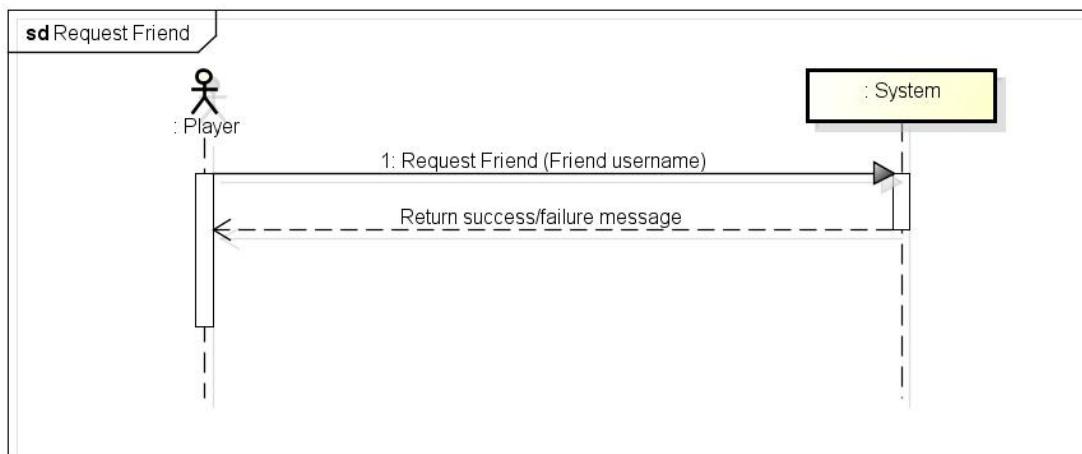


5.3. Manage Friends

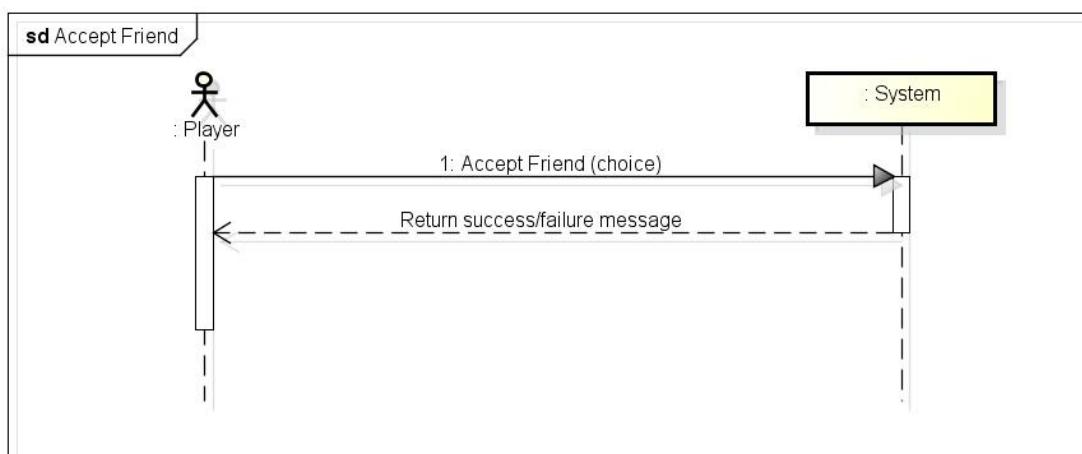
5.3.1. Unfriend Friend



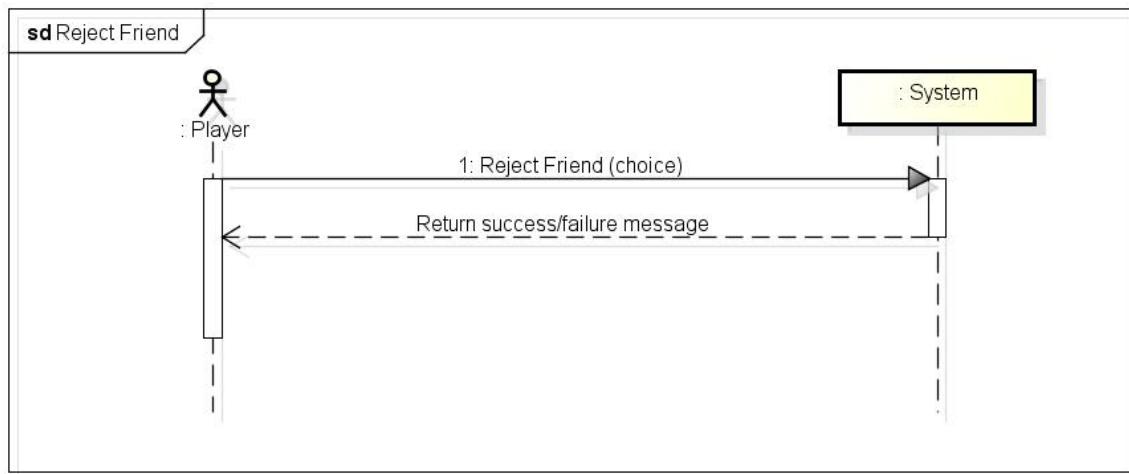
5.3.2. Request Friend



5.3.3. Accept Friend

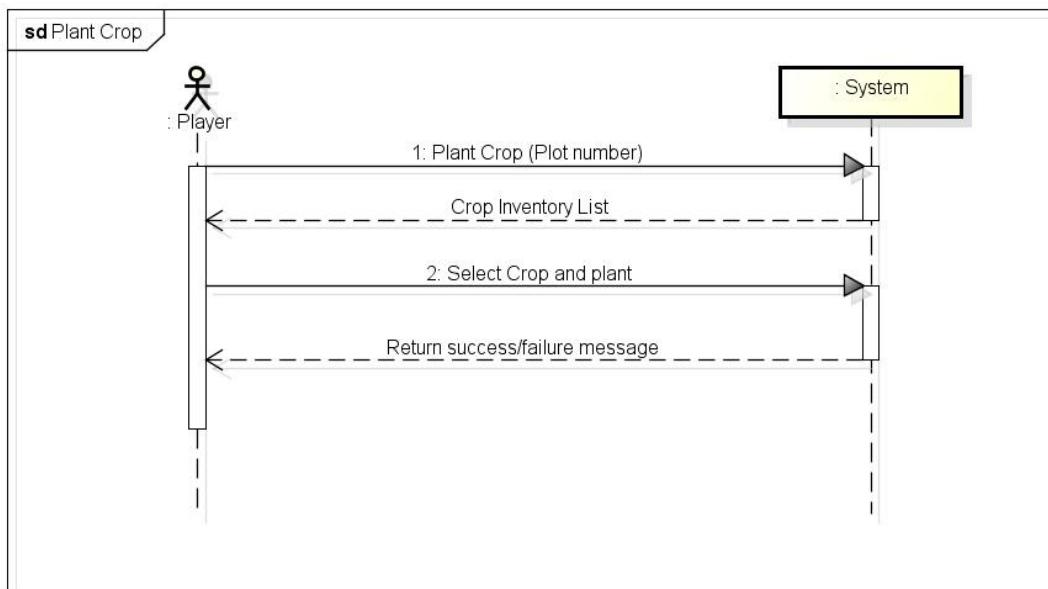


5.3.4. Reject Friend

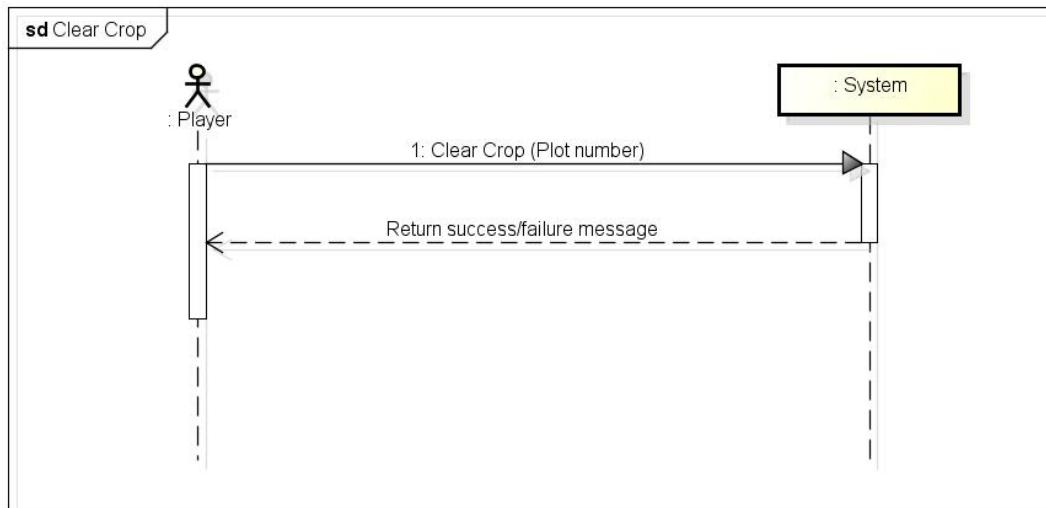


5.4. Manage Farm

5.4.1. Plant Crop



5.4.2. Clear Crop

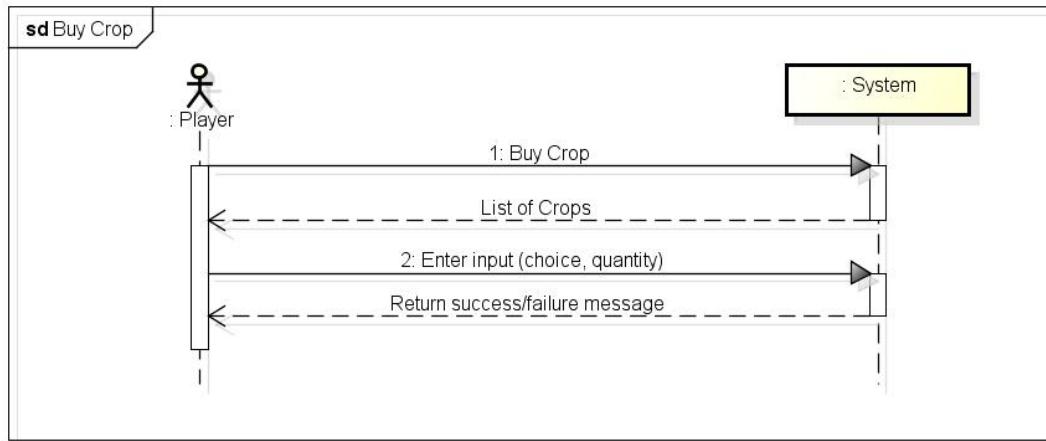


5.4.3. Harvest Crop

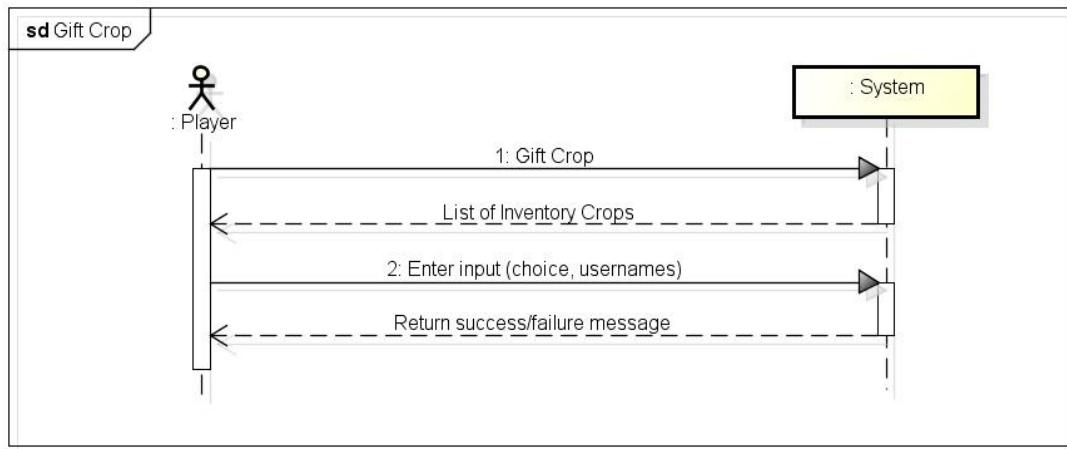


5.5. Manage Inventory

5.5.1. Buy Crop

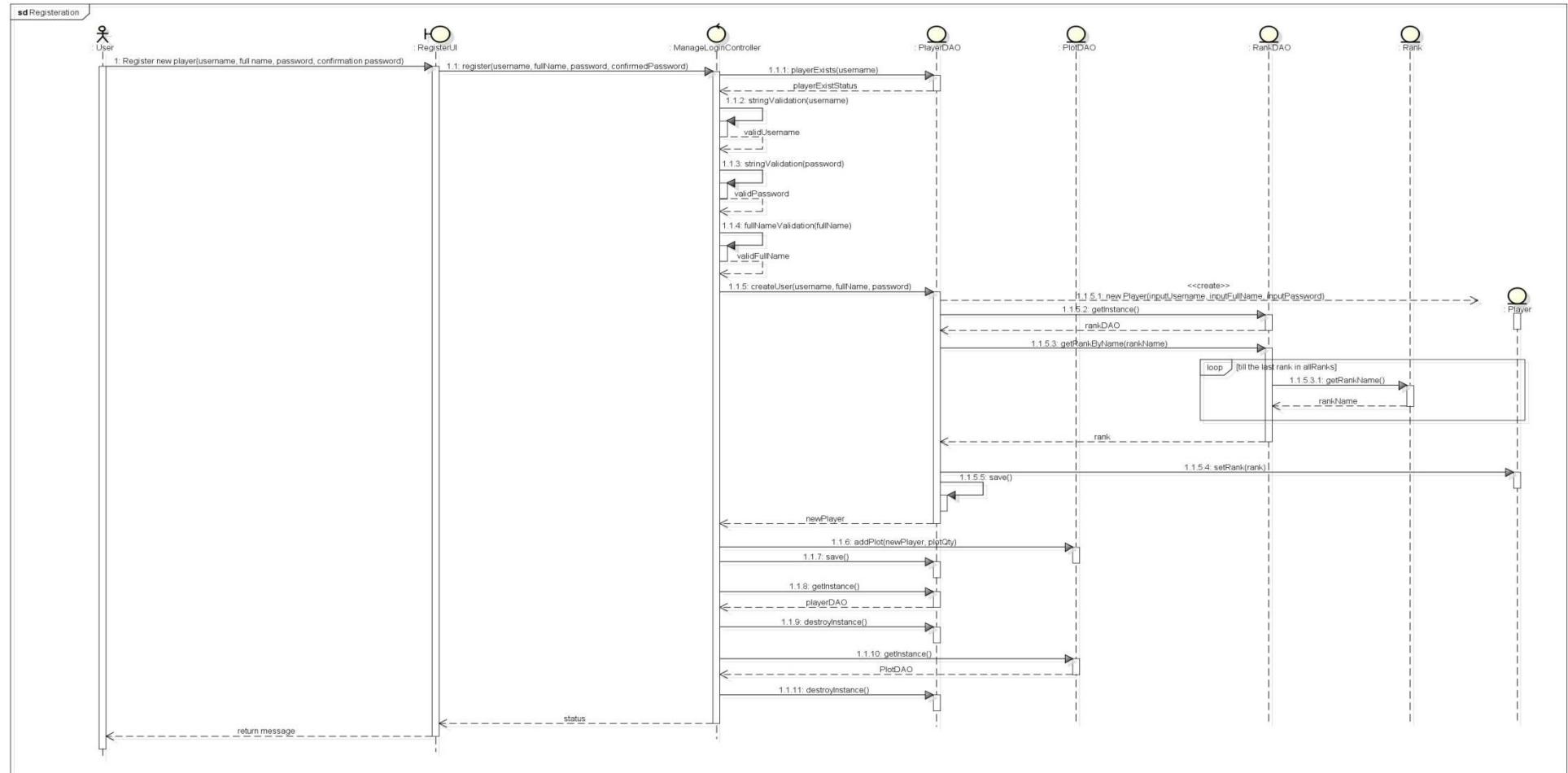


5.5.2. Gift Crop

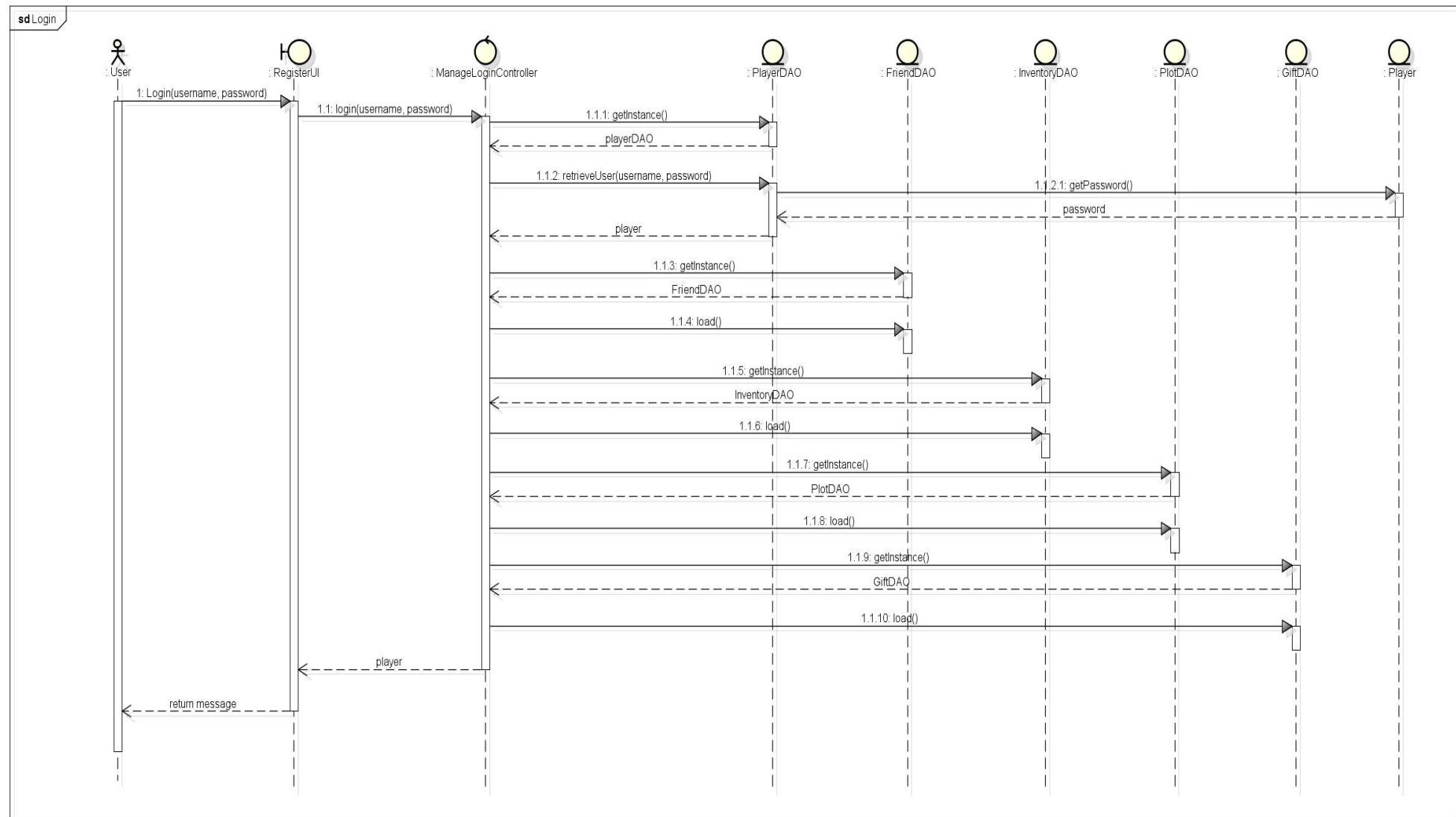


6. Sequence Diagram

6.1. Register

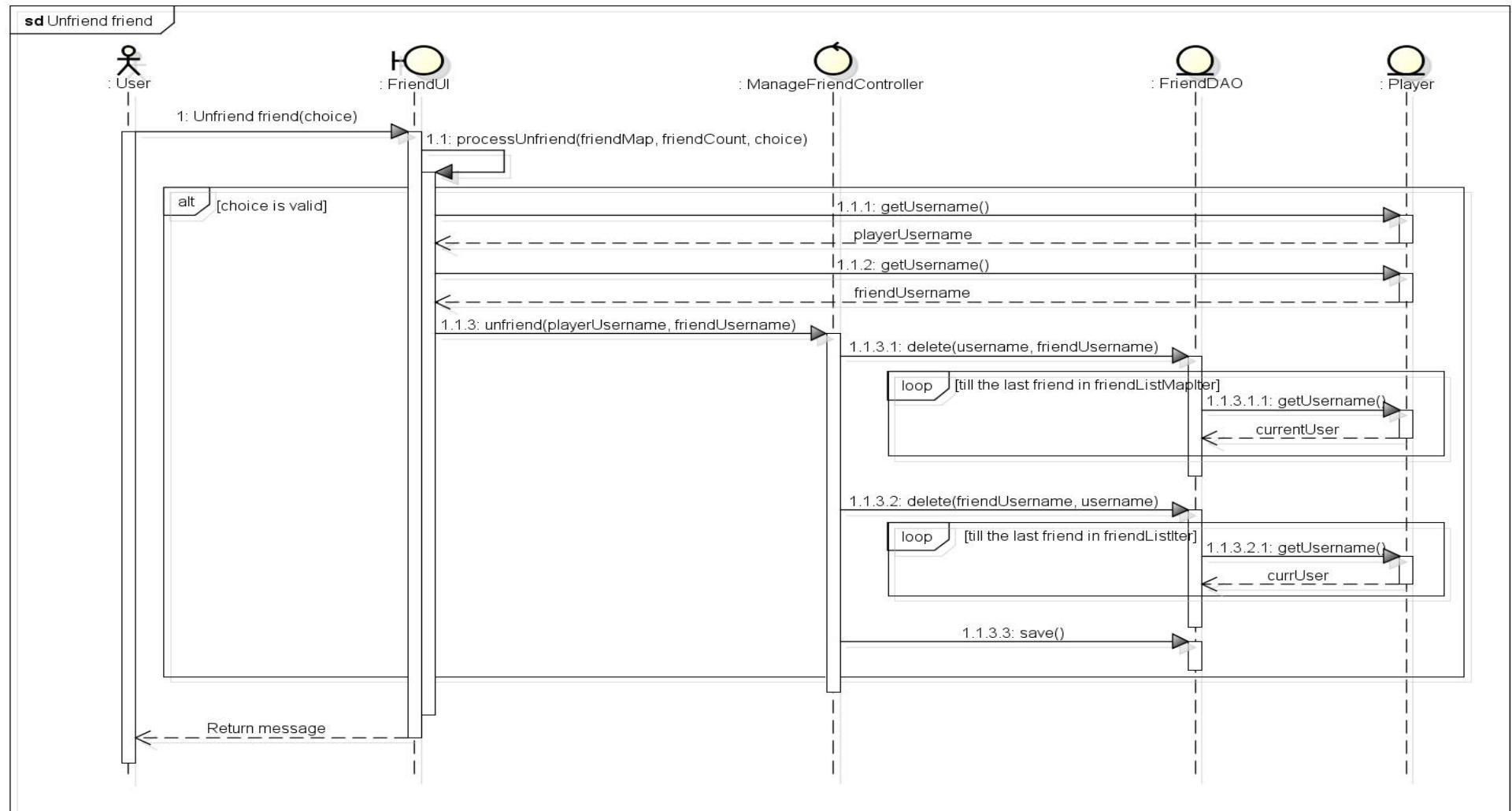


6.2. Login

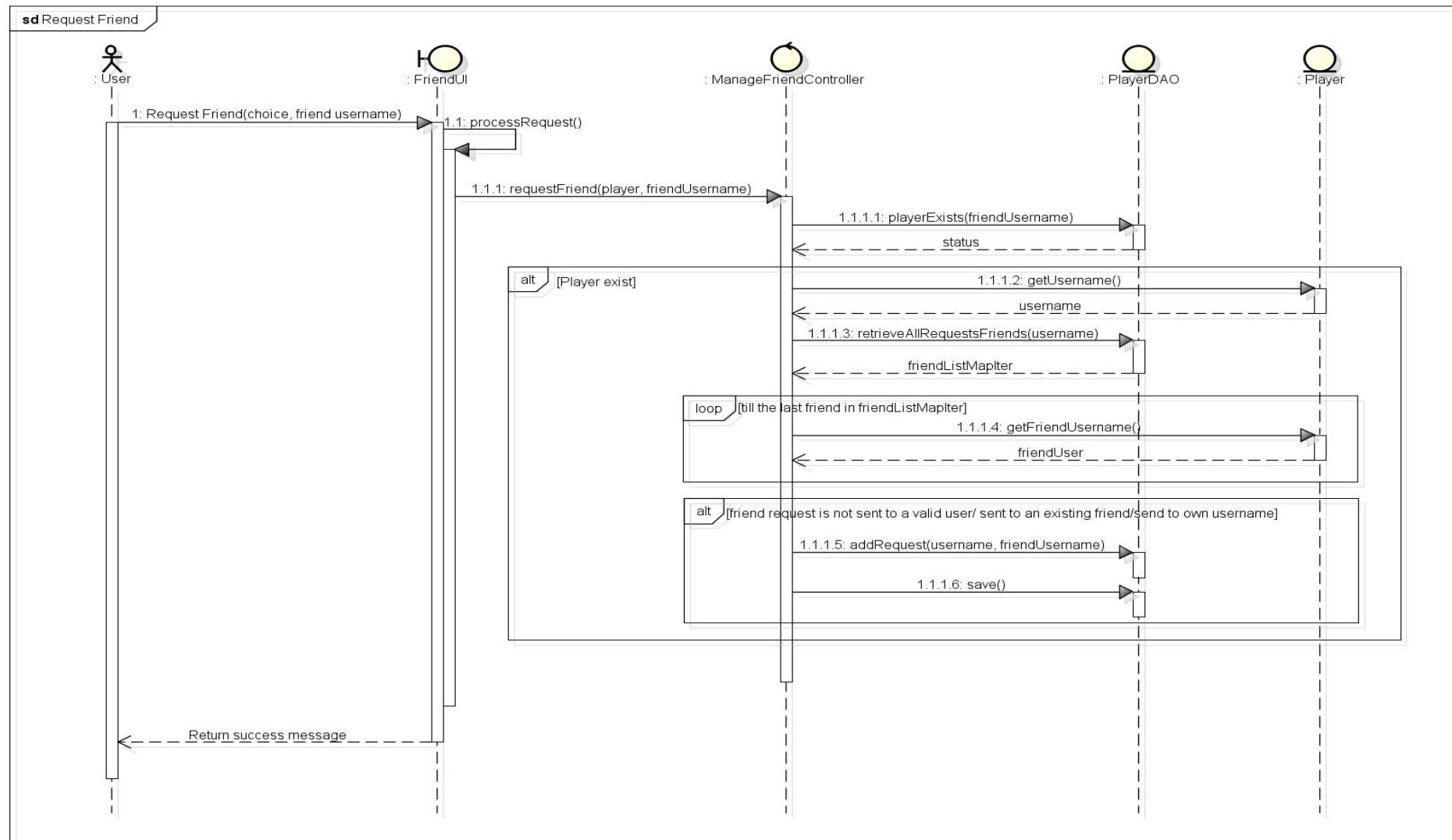


6.3. Manage Friend

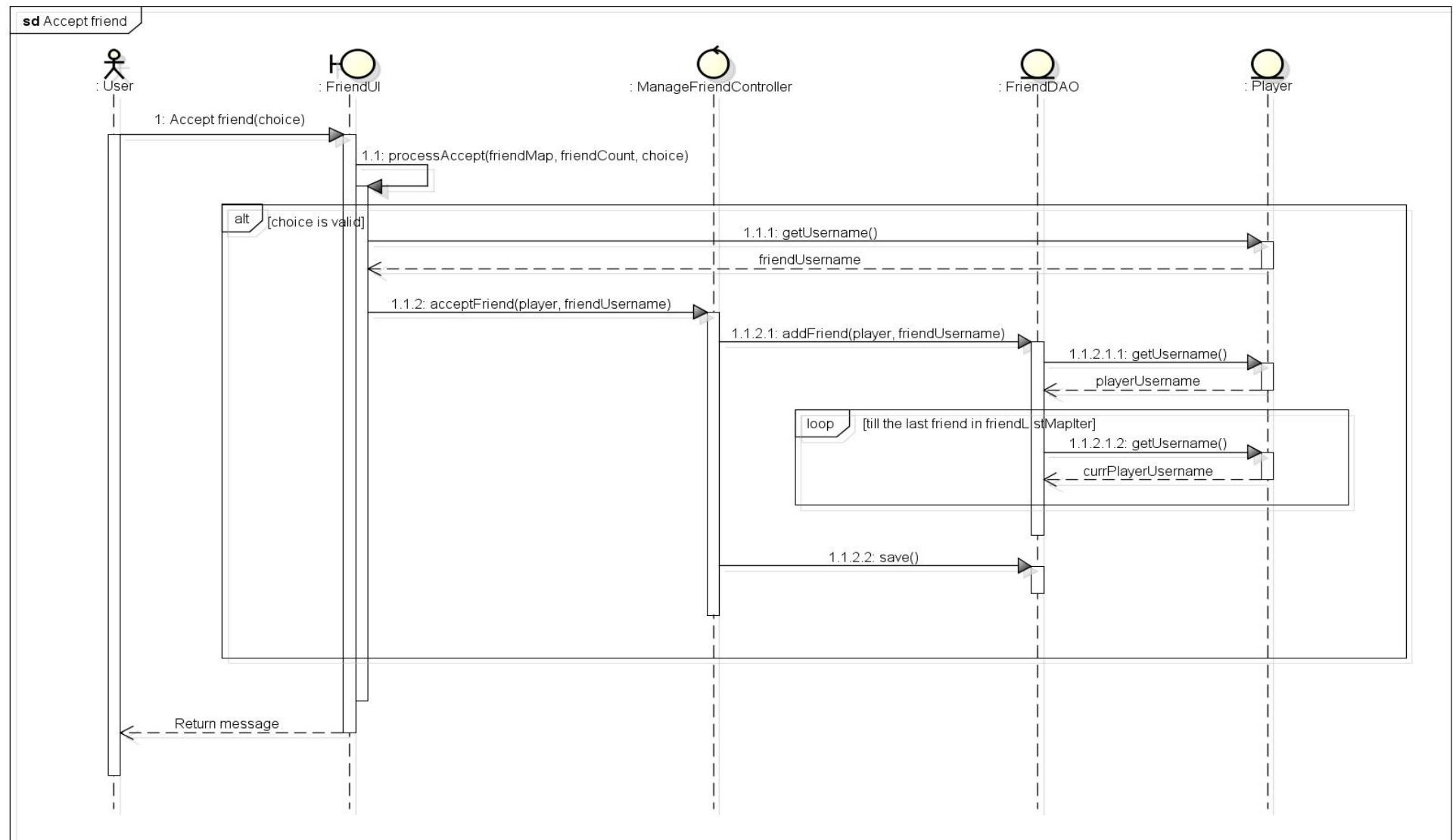
6.3.1. Unfriend Friend



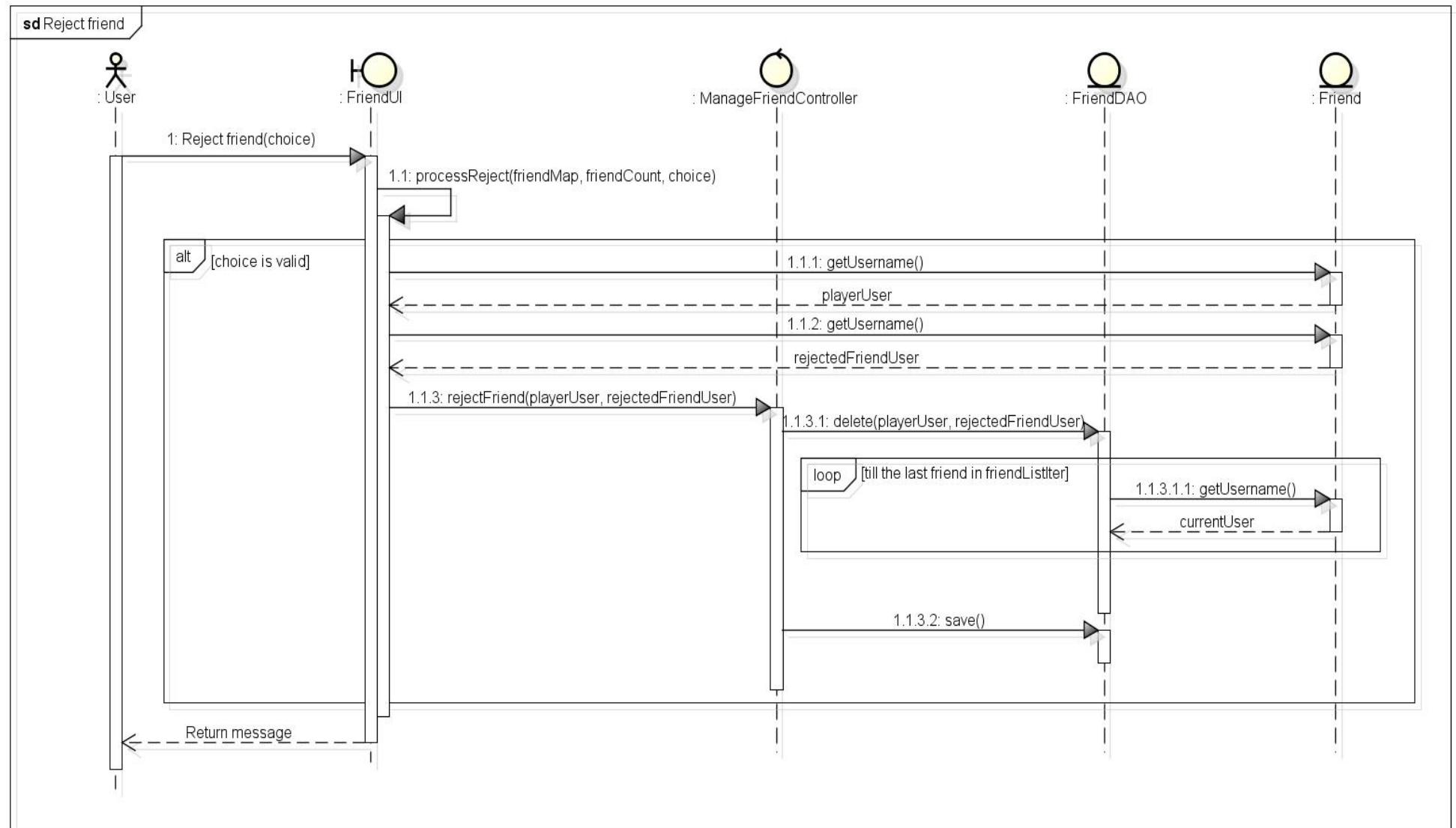
6.3.2. Request Friend



6.3.3. Accept Friend

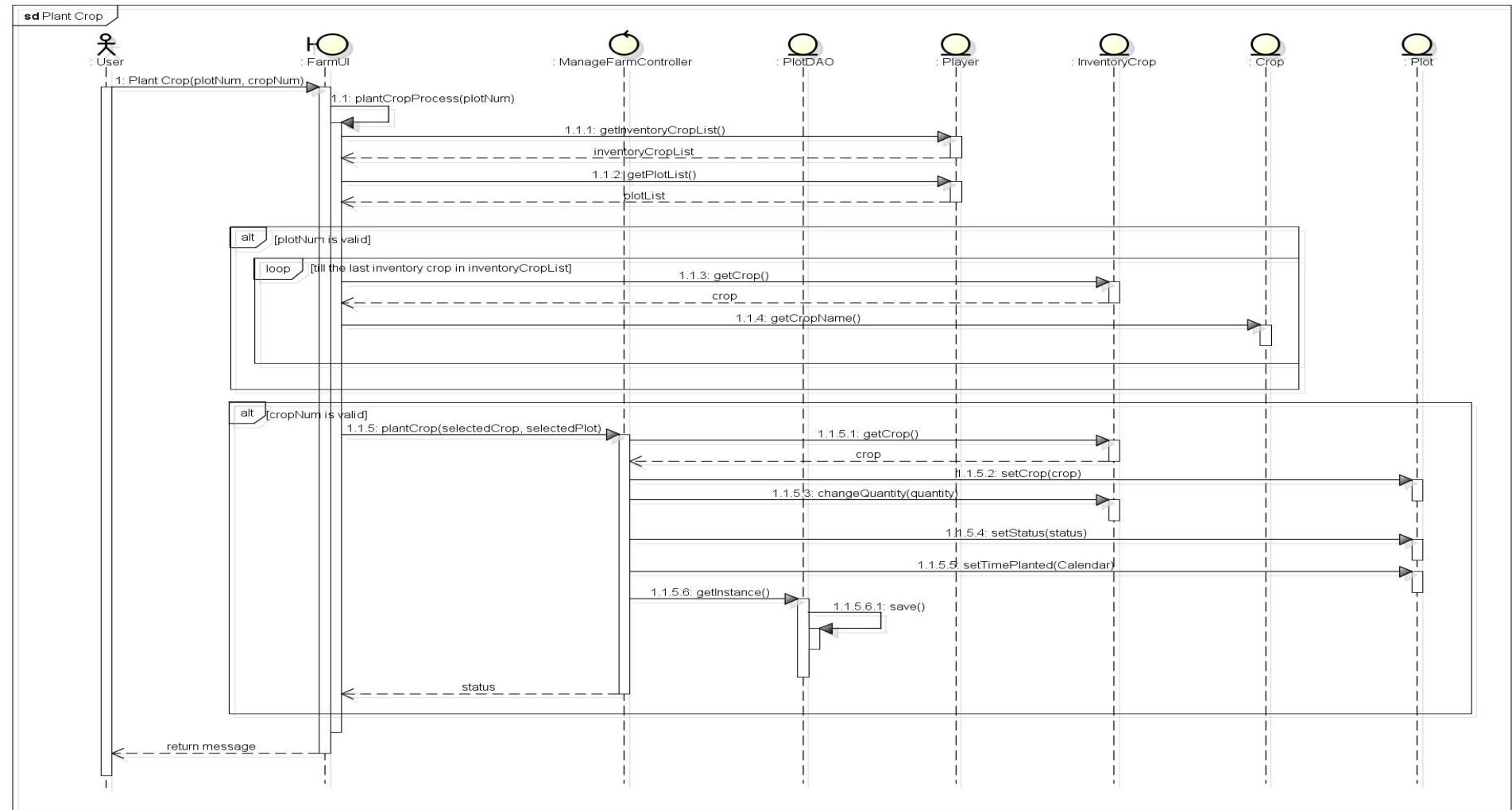


6.3.4. Reject Friend

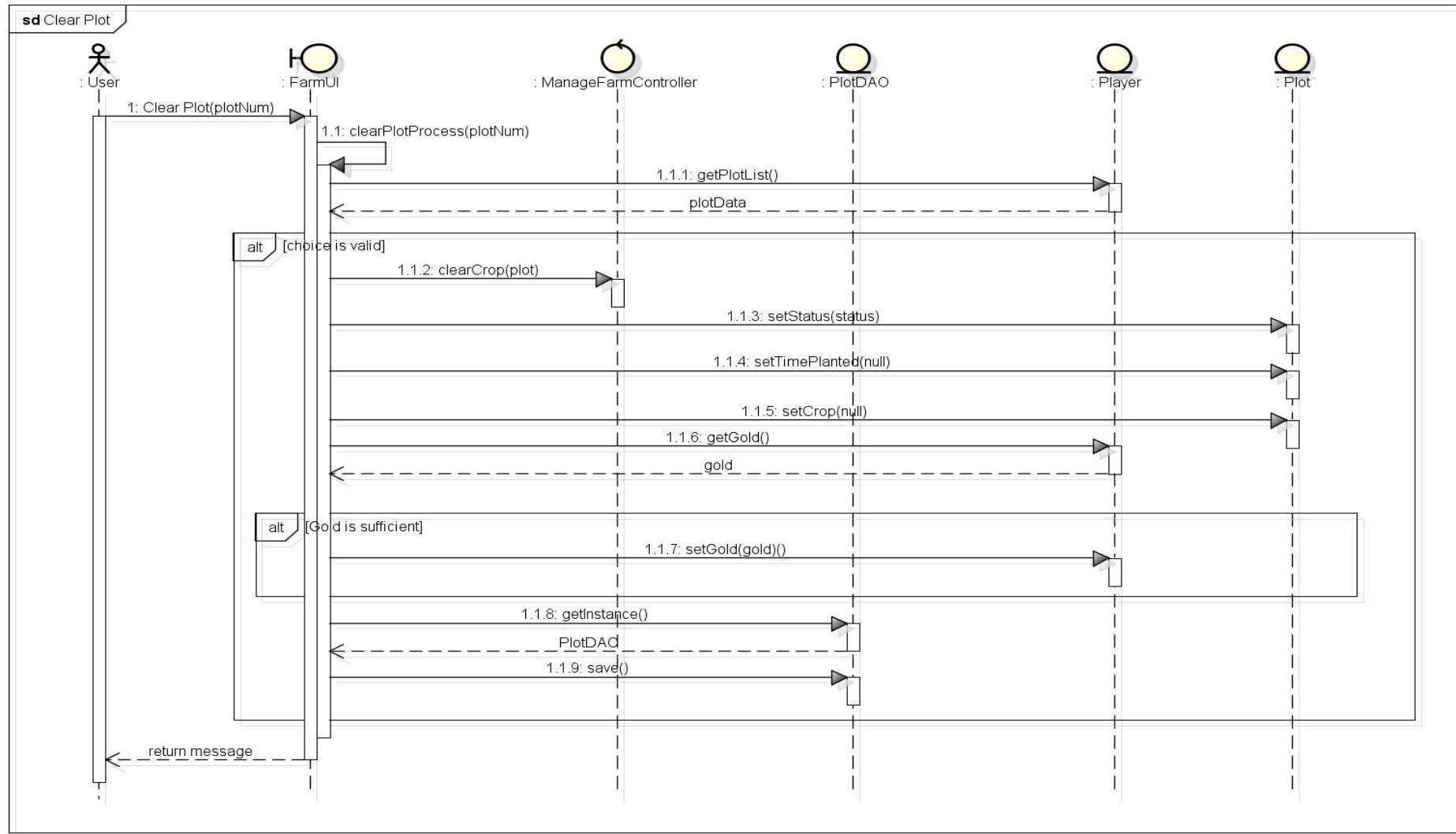


6.4. Manage Farm

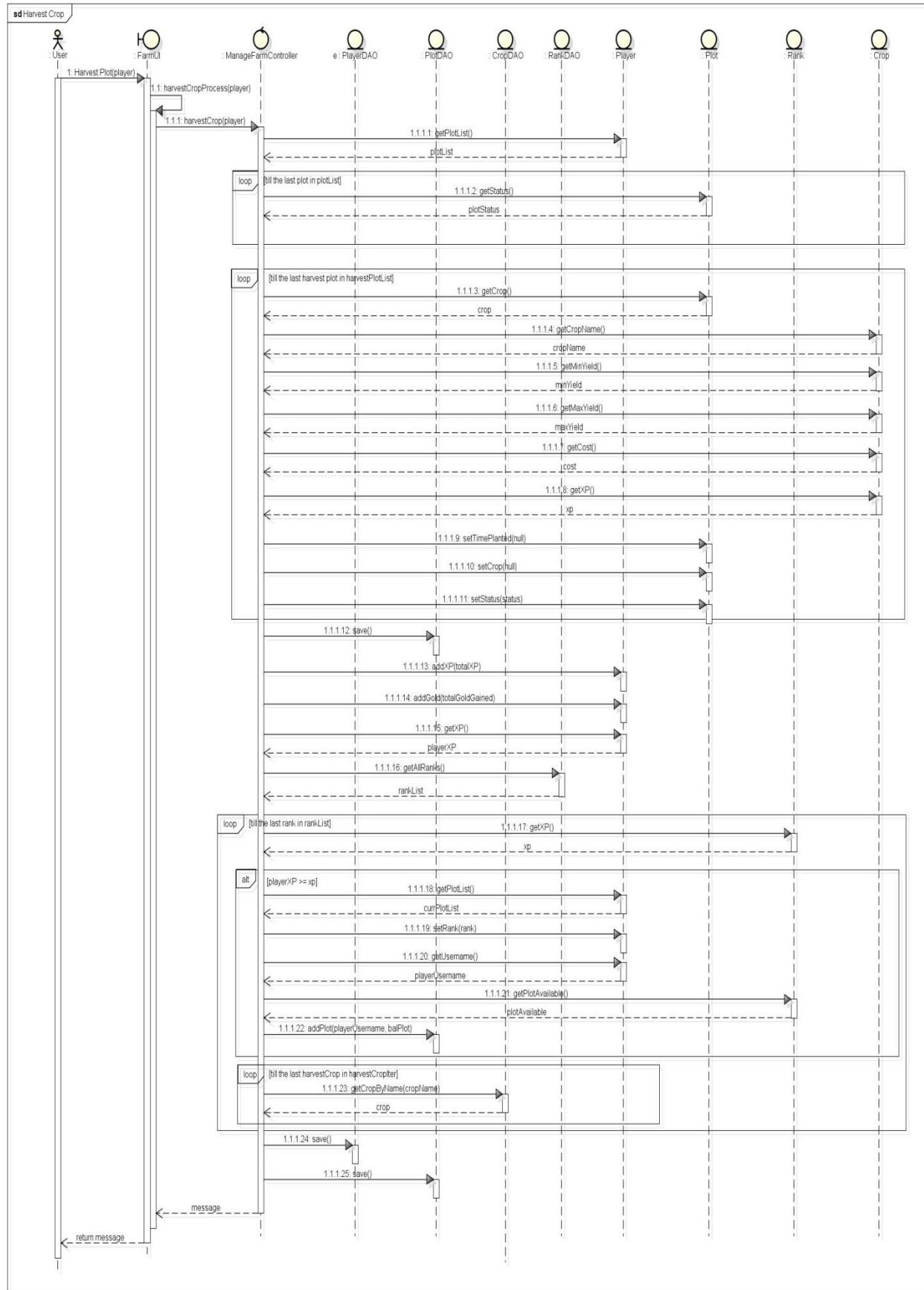
6.4.1. Plant Crop



6.4.2. Clear Crop

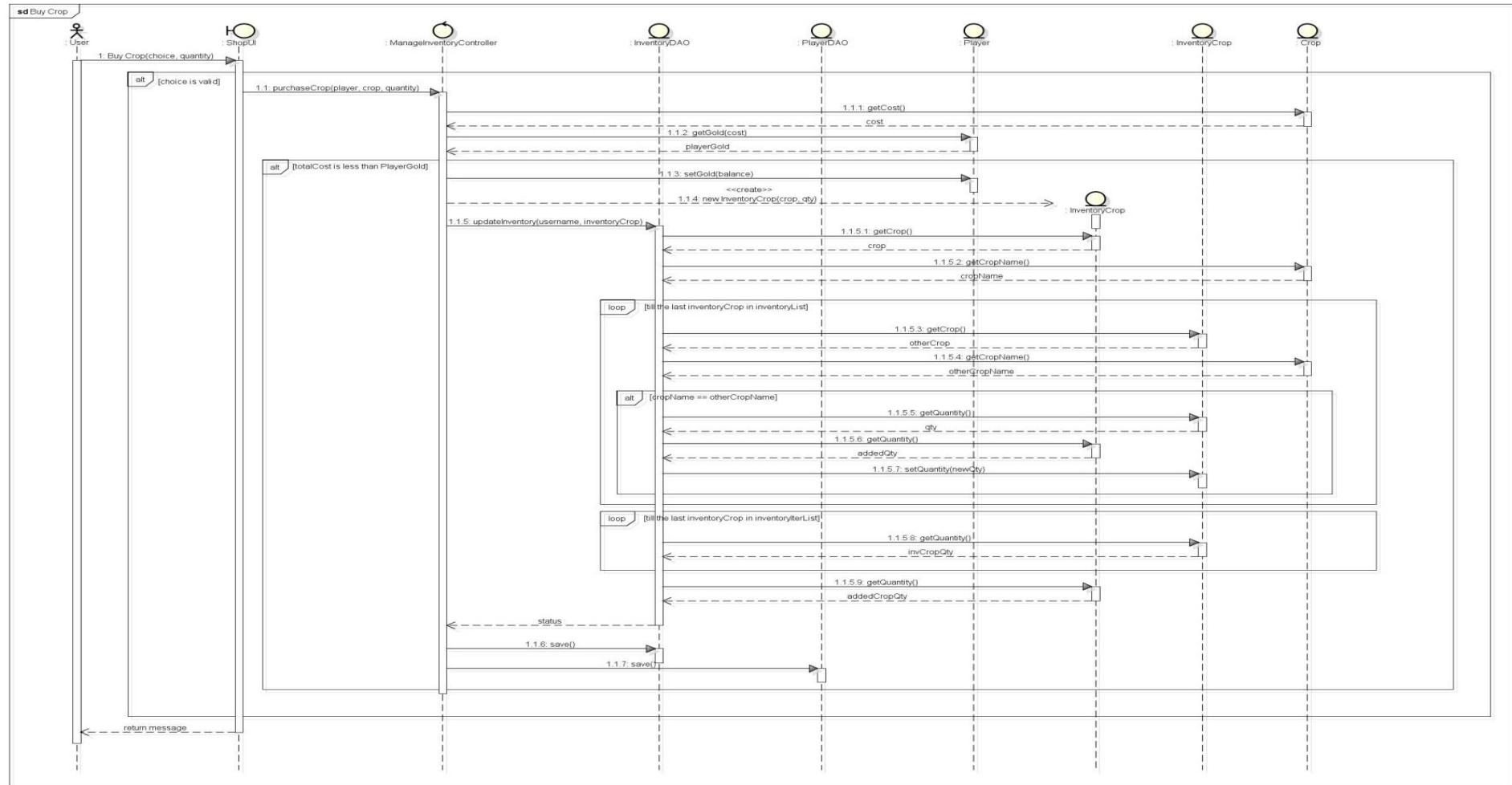


6.4.3. Harvest Crop

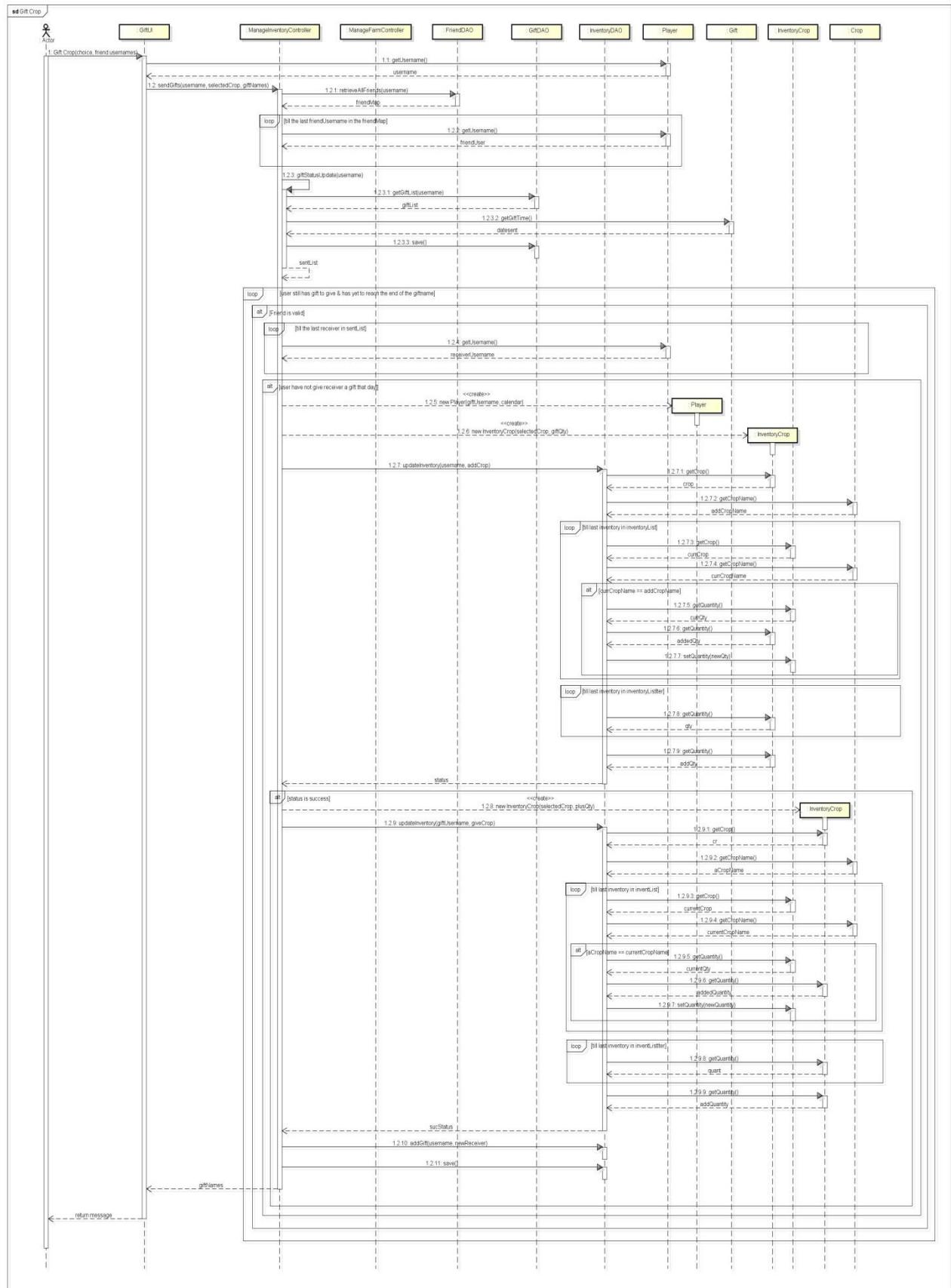


6.5. Manage Inventory

6.5.1. Buy Crop



6.5.2. Gift Crop



7. Design Considerations

With regards to design, our team uses a Boundary, Controller and Entity design. Within our base code, the DAOs being a special class of entities are only allowed direct access by the controllers. This allows for separation of concerns regarding the code and makes it easier to divide and segregate the functionalities for each use case.

When implementing the DAO, our team used a singleton design for all DAOs as it seemed like the best way to handle the DAOs. At any one point in time, it is ensured that only one DAO instance is handling the data and hence ensures that there will not be conflict in data writing or reading.

With regards to handling any potential program crashes due to a lack of necessary files (e.g CSV files that store player data) or possible exceptions within the program, we implemented an ExceptionDAO to fetch exception messages from a file and subsequently throw it to the UI where applicable. Exceptions messages are fetched by its key. This allows for reusability of certain exception messages elsewhere in the code where applicable. However, in the event that the Exception CSV file is missing, the user will be notified of the exception and the program will not run.

8. Challenges faced and lessons learnt

One of the challenges our team faced during this project would be the design of the program. At first, it seemed very unclear on how the design should be like. However, after several iterations and discussions, it became clear on how the design should be like. Ultimately, our controller handles the logic of the program, the UI handles the user display while the DAO handles only file reading and writing.

Another challenge that we faced was user input validation. Input validation was a major obstacle as it was hard to predict what users would enter into the program. As such, to ensure that the program would not crash due to exceptions such as NullPointerException. Vigorous testing

Time management is also an issue. However, we still found time to meet up weekly so as to ensure that our project's progress does not lack behind. Consistency is really key when it comes to this project as last minute work would not serve much good.

9. Appendix

