

[IS201] Object Oriented Application Development

Farm City – Final Project Submission

[G3-T02] WANG SHUWEI | TAN YONG MING ALVIN | LIM JIA KAI

INSTRUCTOR: PROF David Lo

April 7, 2015

PROJECT - Farm City

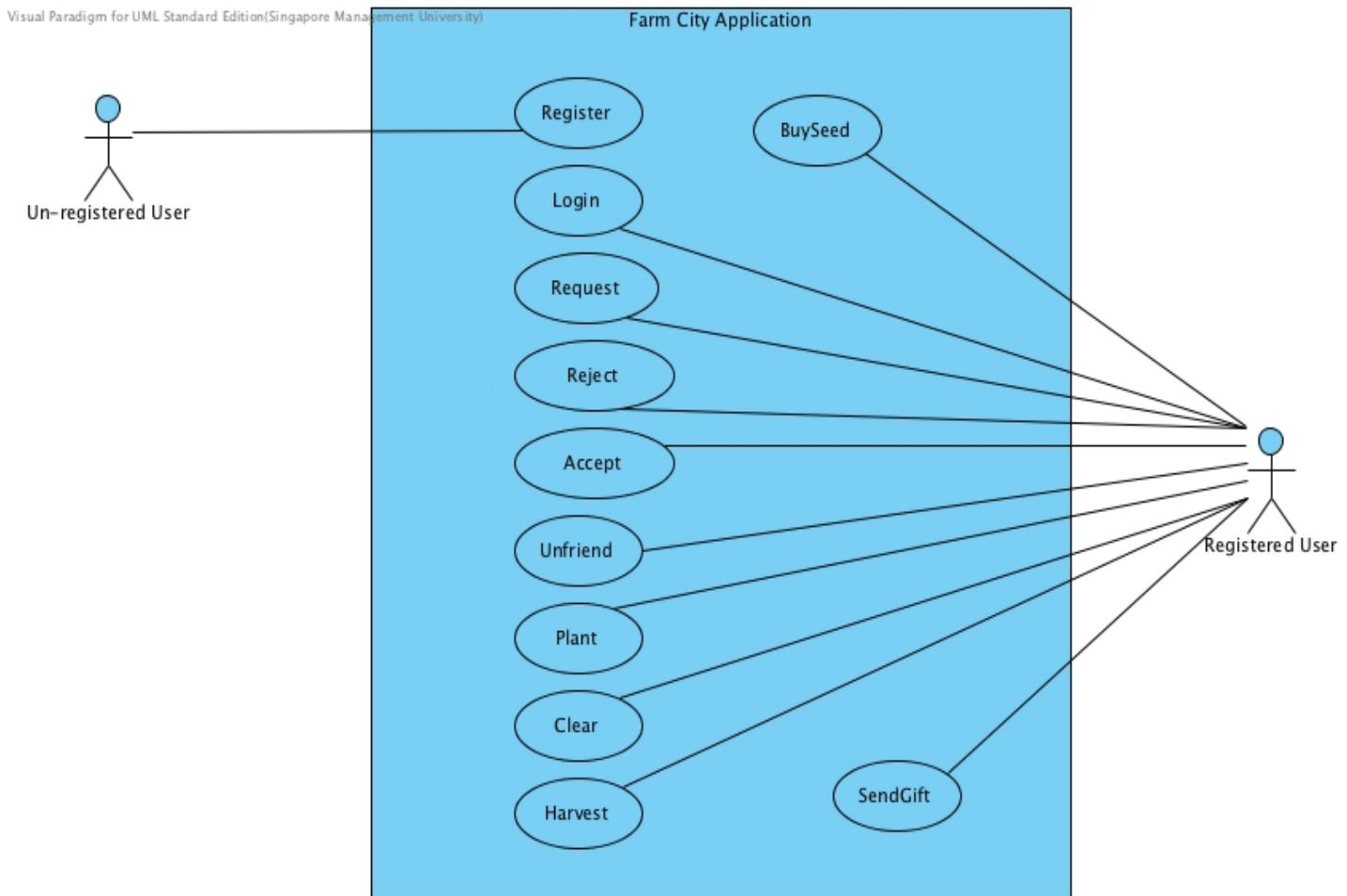
Table of Contents

1 ANALYSIS & DESIGN	3
1.1 Use Case Diagram	3
1.2 Domain Diagram	5
1.3 Class Diagram.....	6
1.4 Use Case.....	7
1.4.1 <i>Register</i>	7
1.4.2 <i>Login</i>	10
1.4.3 <i>Send Friend Request</i>	13
1.4.4 <i>Accept Friend Request</i>	16
1.4.5 <i>Reject Friend Request</i>	19
1.4.6 <i>Unfriend</i>	21
1.4.7 <i>Plant</i>	23
1.4.8 <i>Harvest</i>	26
1.4.9 <i>Clear</i>	28
1.4.10 <i>Buy Seeds</i>	31
1.4.11 <i>Send Gift</i>	34
1.5 Object Oriented Design Consideration	37
2 MISCELLANEOUS	38
2.1 The Challenge.....	38
2.2 What we have learnt.....	38
3 SCREENSHOTS OF SAMPLE RUNS.....	39
3.1 Register	39
3.2 Login.....	41
3.3 Send Friend Request	42
3.4 Accept Friend Request	46
3.5 Reject Friend Request	48
3.6 Unfriend	49
3.7 Plant	51
3.8 Harvest	53
3.9 Clear	55
3.10 Buy Seeds.....	57
3.11 Send Gift	59

PROJECT - Farm City

1 ANALYSIS & DESIGN

1.1 Use Case Diagram



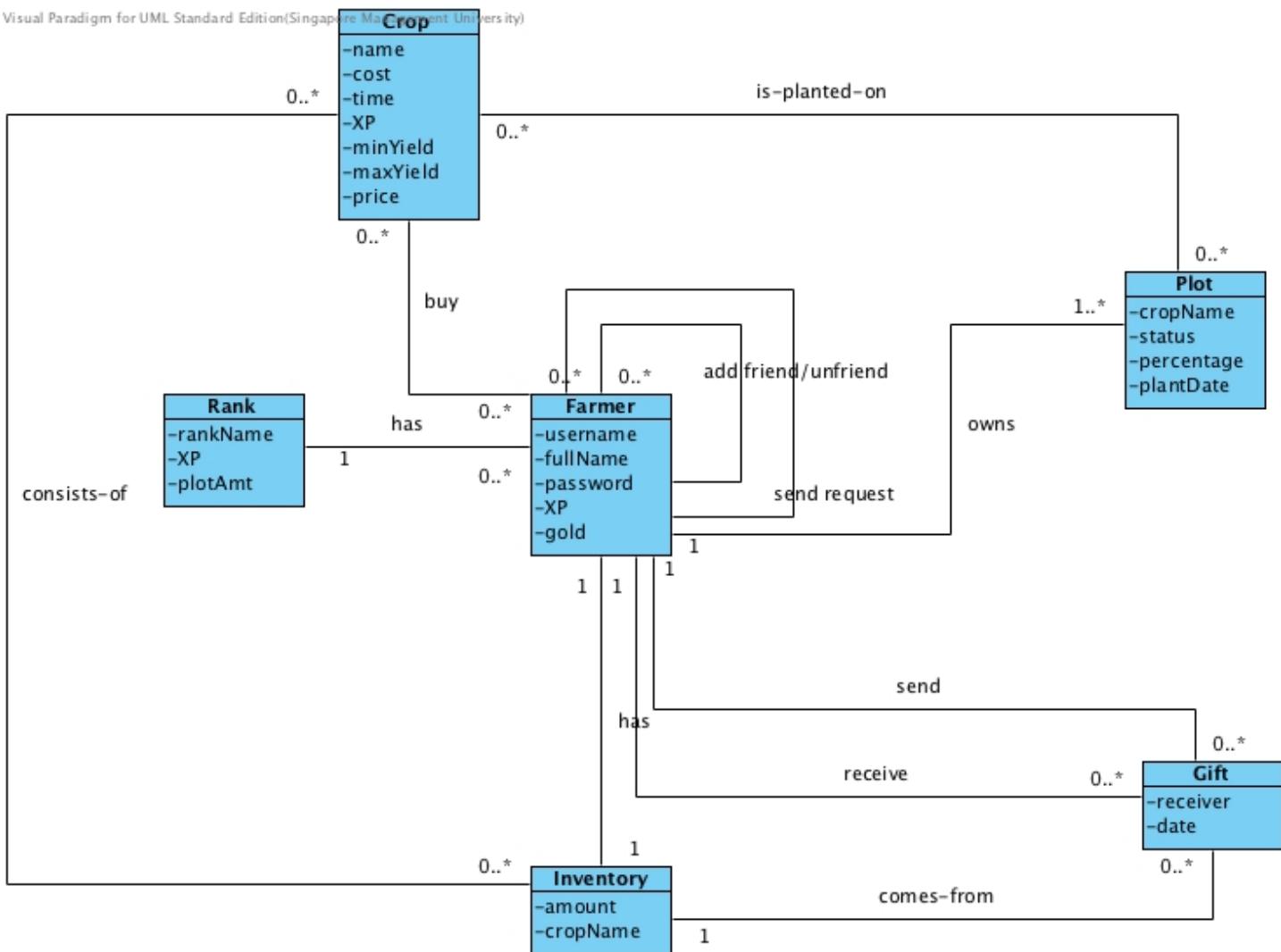
PROJECT - Farm City

Use Case	Description
Register	This use case allows the player to create a new account.
Login	This use case allows the player to login with a registered username and password.
Request	This use case allows the player to send a friend request to another player.
Accept	This use case allows the player to accept a friend request from another player.
Reject	This use case allows the player to reject a friend request from another player.
Unfriend	This use case allows the player to delete a friend from the friend list.
Plant	The use case allows the player to plant the crop on the plot.
Harvest	The use case allows the player to harvest the crop.
Clear	The use case allows the player to clear the wilted crop.
Buy Seeds	This use case allows the player to buy seed from store.
Send Gift	This use case allows the player to send gift to friends.

PROJECT - Farm City

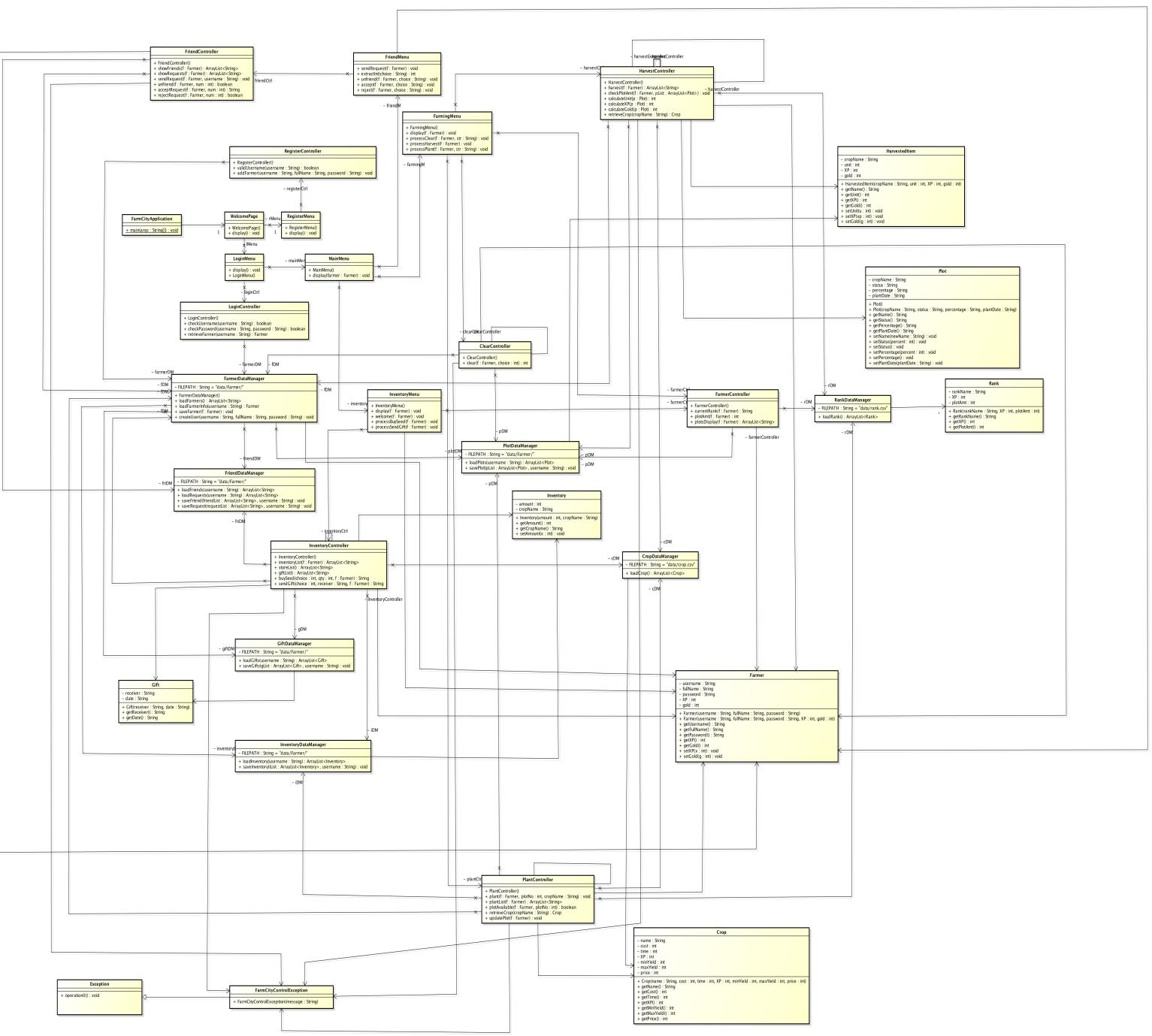
1.2 Domain Diagram

Visual Paradigm for UML Standard Edition(Singapore Management University)



PROJECT - Farm City

1.3 Class Diagram



PROJECT - Farm City

1.4 Use Case

1.4.1 Register

1.4.1.1 *Use Case Scenario*

Actor: Unregistered user

Precondition: The user is not logged in.

Main flow of events:

1. The use case begins when an unregistered user choose to register
2. The system prompts the user to enter a username
3. The system prompts the user to enter his/her full name and password
4. The system prompts the user to enter his/her password again to confirm
5. The system informs the user that a farmer account has been created
6. Use case ends.

Alternate flows:

2a. The username entered is not alphanumeric

- i. The system informs user that username is not alphanumeric
- ii. Use case continues in main flow 2.

2b. The username entered is not unique

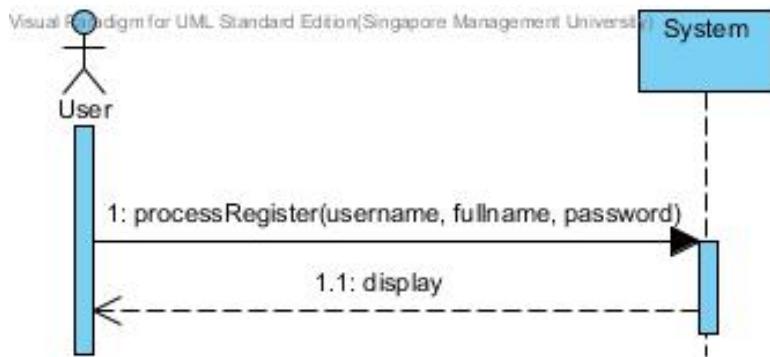
- i. The system informs user that username is not unique
- ii. Use case continues in main flow 2.

4a. The confirm password entered is different from the first password entered

- i. The system informs that the confirmed password does not match
- ii. Use case continues in main flow 4.

PROJECT - Farm City

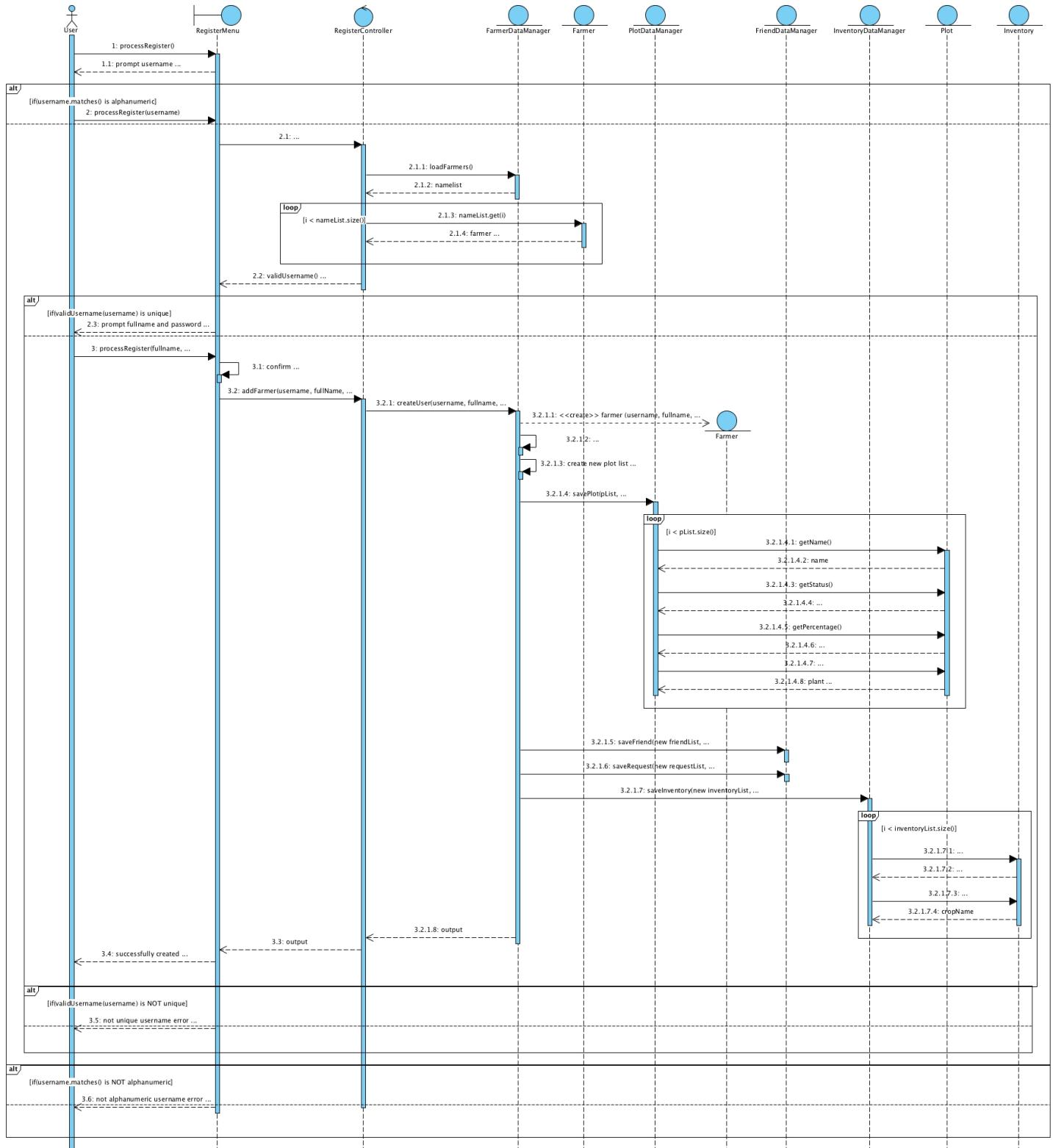
1.4.1.2 System Sequence Diagram



1.4.1.3 Sequence Diagram

PROJECT - Farm City

UML Paradigm for UML Standard Edition (Singapore Management University)



PROJECT - Farm City

1.4.2 Login

1.4.2.1 *Use Case Scenario*

Actor: Farmer

Precondition: The user is not logged in.

Main flow of events:

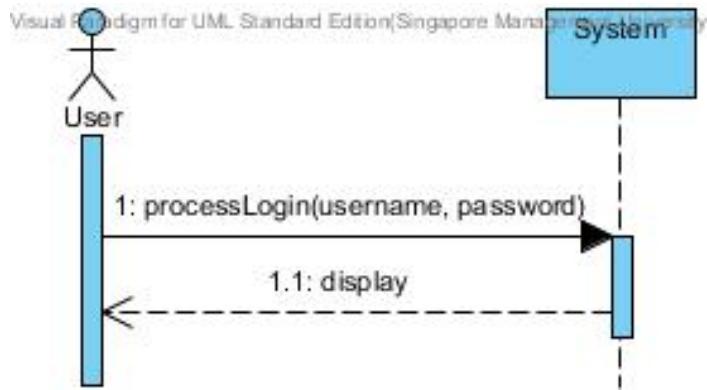
7. The use case begins when a farmer choose to login
8. The system prompts the user to enter a username
9. The system prompts the user to enter a password
10. The system inform the user login has been successful and display the logon page
11. Use case ends.

Alternate flows:

- 3a. The username entered does not exist
 - iii. The system informs user that username does not exist
 - iv. The case ends.
- 4a. The password does not match the username
 - iii. The system informs user that the password is word
 - iv. Use case ends.

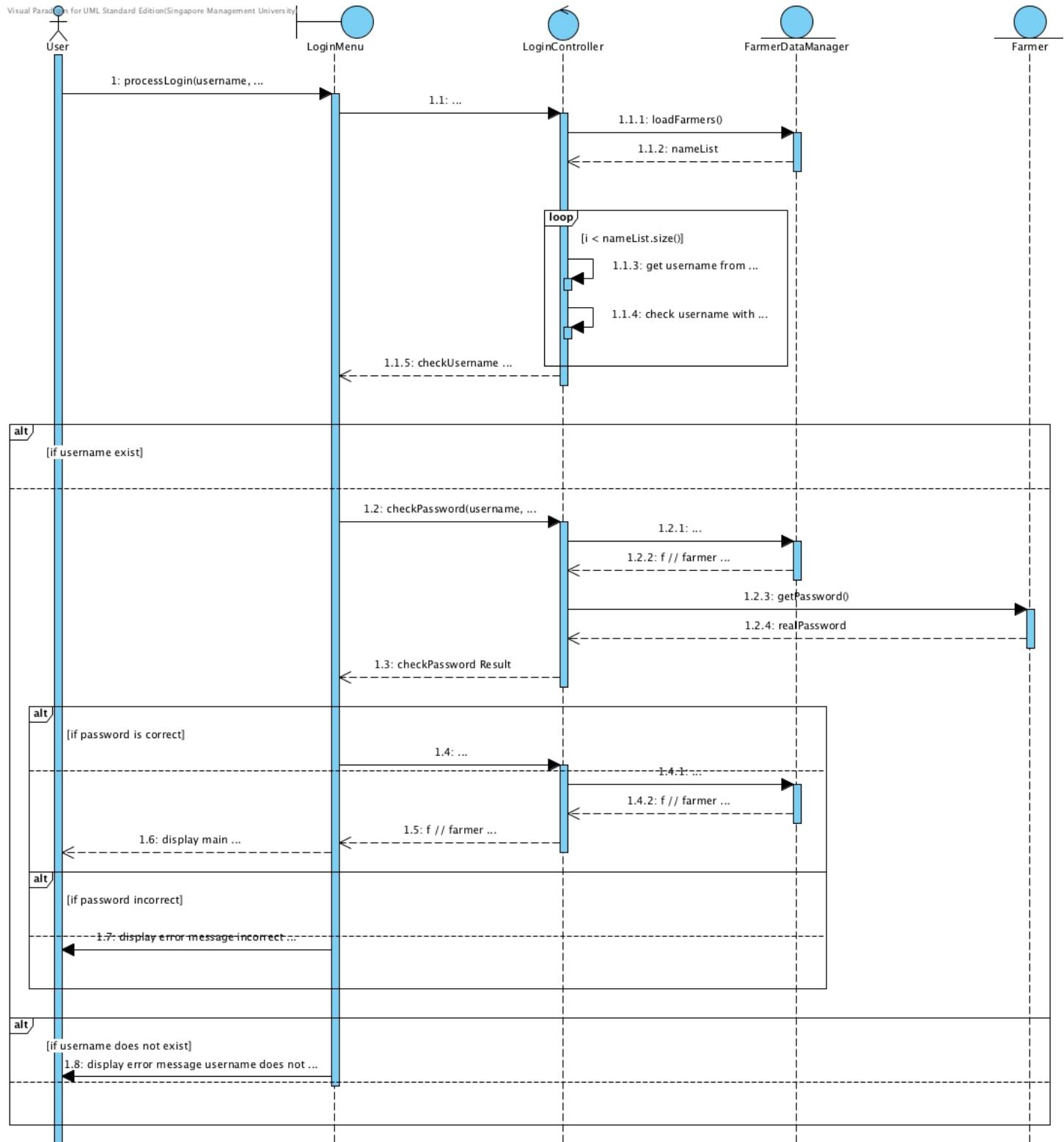
PROJECT - Farm City

1.4.2.2 System Sequence Diagram



1.4.2.3 Sequence Diagram

PROJECT - Farm City



PROJECT - Farm City

1.4.3 Send Friend Request

1.4.3.1 *Use Case Scenario*

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

12. The use case begins when farmer chooses to send a friend request
13. The system prompt the user to enter the username of that friend
14. The farmer enter a username
15. The system check if farmer is already friends with username
16. The system check if farmer is trying to send a request to himself/herself
17. Use case ends.

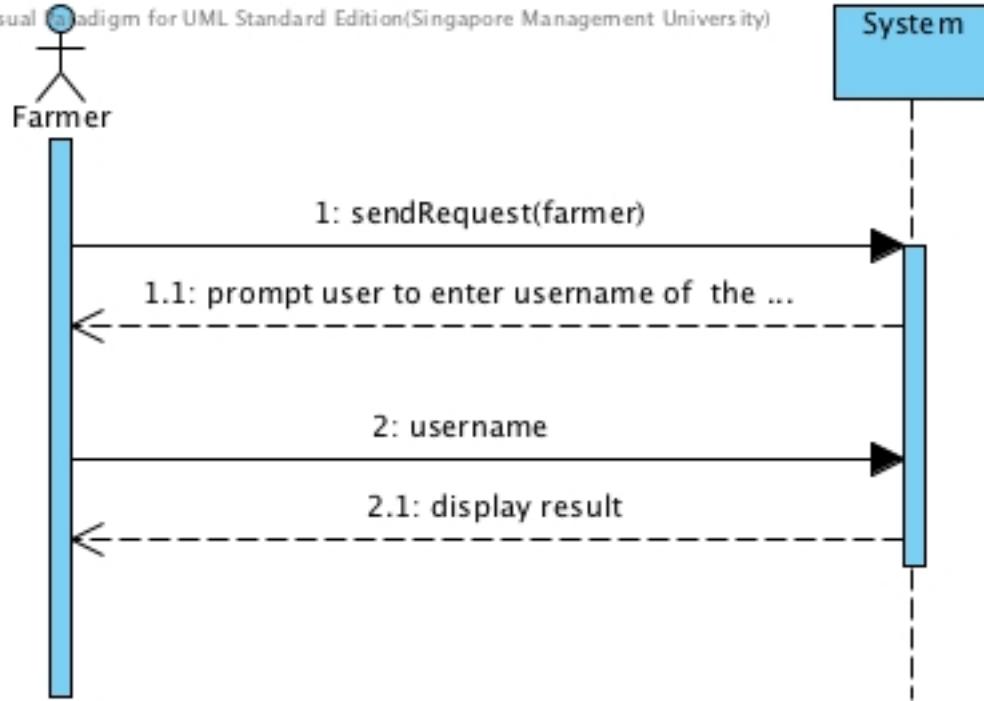
Alternate flows:

- 3a. The user is trying to send a friend request to a username that he/she had sent before
 - i. The system informs farmer that he/she is trying to send a friend request to a username that he/she had sent before
 - ii. The use case ends.
- 3b. The user is trying to send a friend request to a username that does not exist
 - i. The system informs farmer that the username does not exist
 - ii. The use case ends.
- 3c. The farmer is trying to send a friend request to himself/herself
 - v. The system informs farmer that he/she is trying to send a friend request to himself/herself
 - vi. The use case ends.
- 3d. The farmer is already friends with username
 - v. The system informs farmer that he/she is already friends with username
 - vi. The use case ends.

PROJECT - Farm City

1.4.3.2 System Sequence Diagram

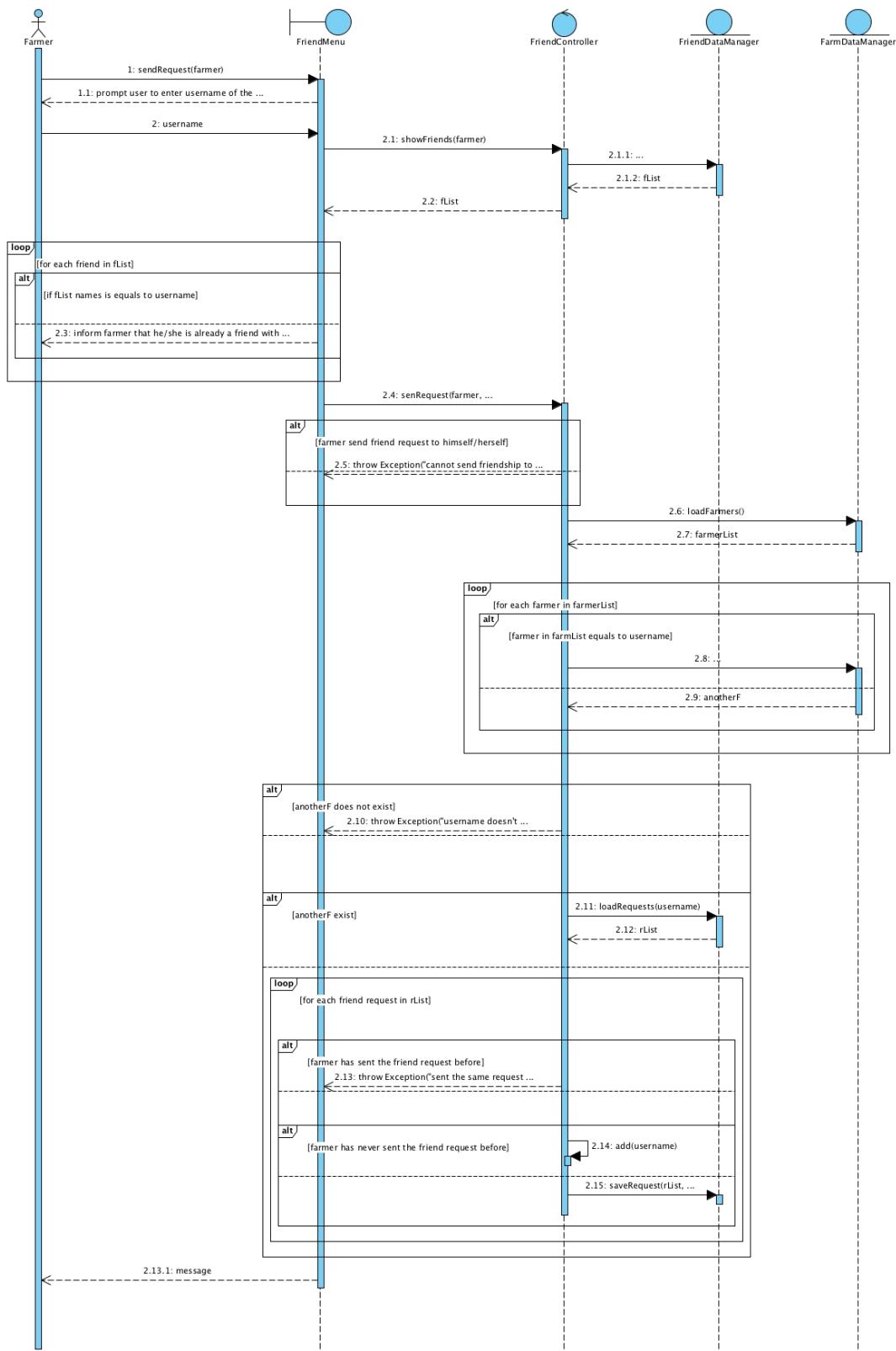
Visual Paradigm for UML Standard Edition(Singapore Management University)



1.4.3.3 Sequence Diagram

PROJECT - Farm City

UML Paradigm for UML Standard Edition(Singapore Management University)



PROJECT - Farm City

1.4.4 Accept Friend Request

1.4.4.1 *Use Case Scenario*

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

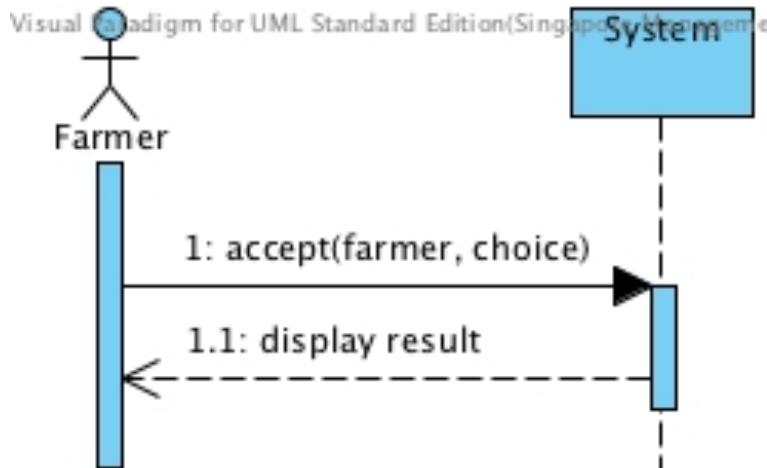
18. The use case begins when farmer chooses to accept a friend request
19. The system check if the friend requestor is already the farmer's friend
20. The system check if the farmer exist
21. The system adds the friend request into the farmer's and the friend requestor's friend list
22. Use case ends.

Alternate flows:

- 2a. The friend requestor is already friends with the farmer
 - vii. The system informs farmer that the friend requestor is already friends with the farmer
 - viii. The use case ends.
- 3a. The friend requestor does not exist
 - vii. The system informs farmer that the friend requestor does not exist
 - viii. The use case ends

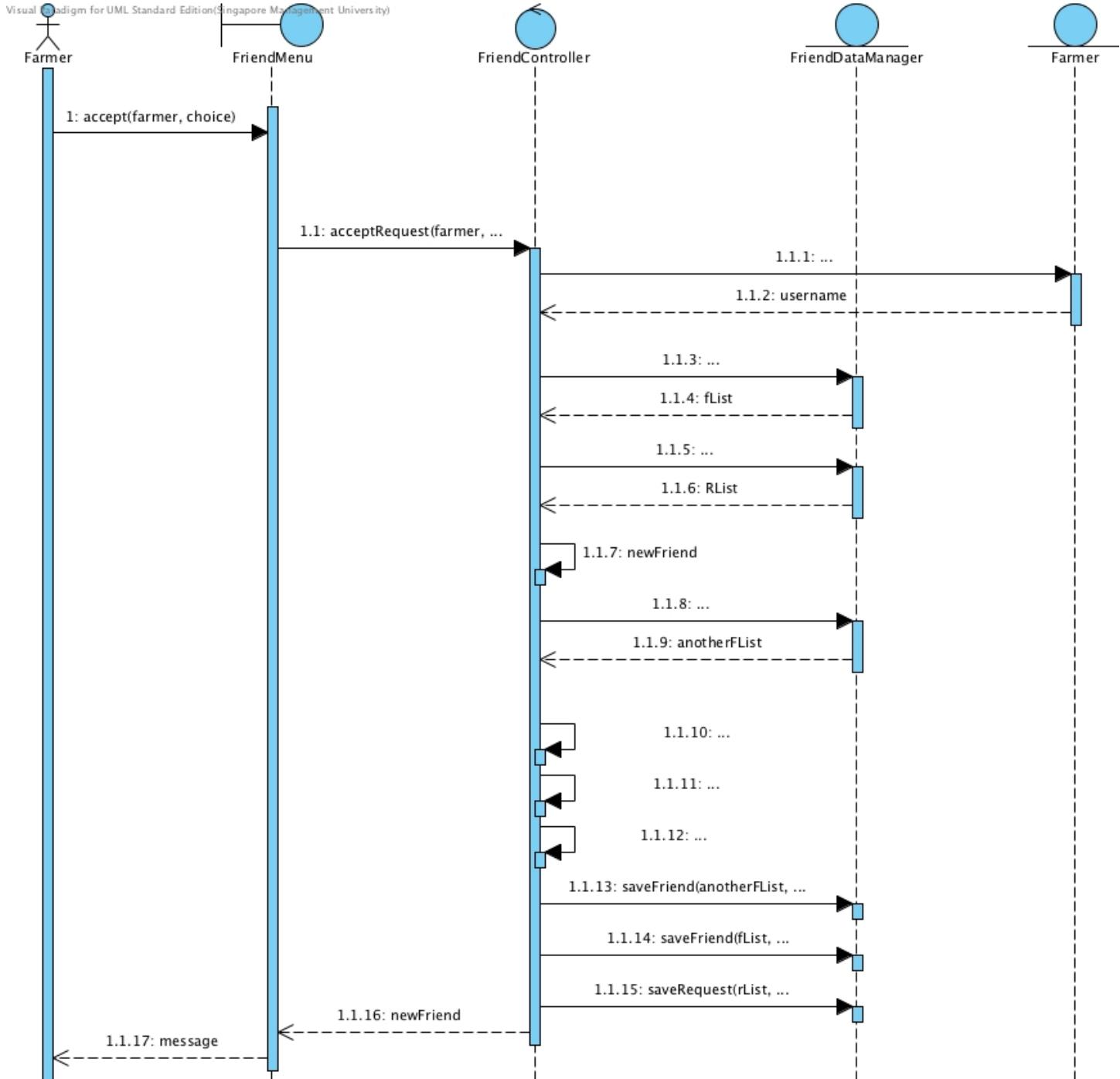
PROJECT - Farm City

1.4.4.2 System Sequence Diagram



PROJECT - Farm City

1.4.4.3 Sequence Diagram



PROJECT - Farm City

1.4.5 Reject Friend Request

1.4.5.1 Use Case Scenario

Actor: Farmer

Precondition: The farmer is logged in.

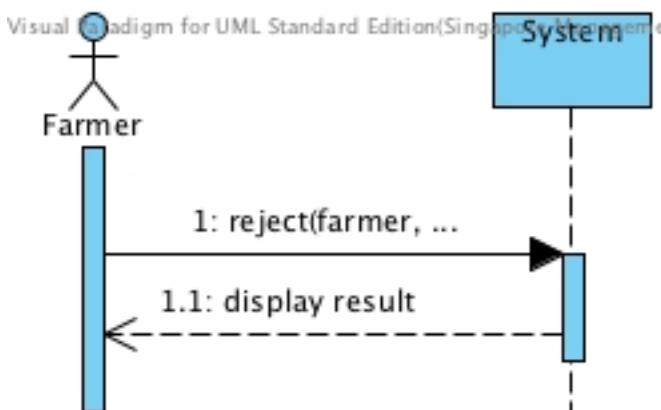
Main flow of events:

23. The use case begins when farmer chooses to request a friend request
24. The farmer enters the request he/she wants to reject
25. The system remove the friend question from the request list
26. Use case ends.

Alternate flows:

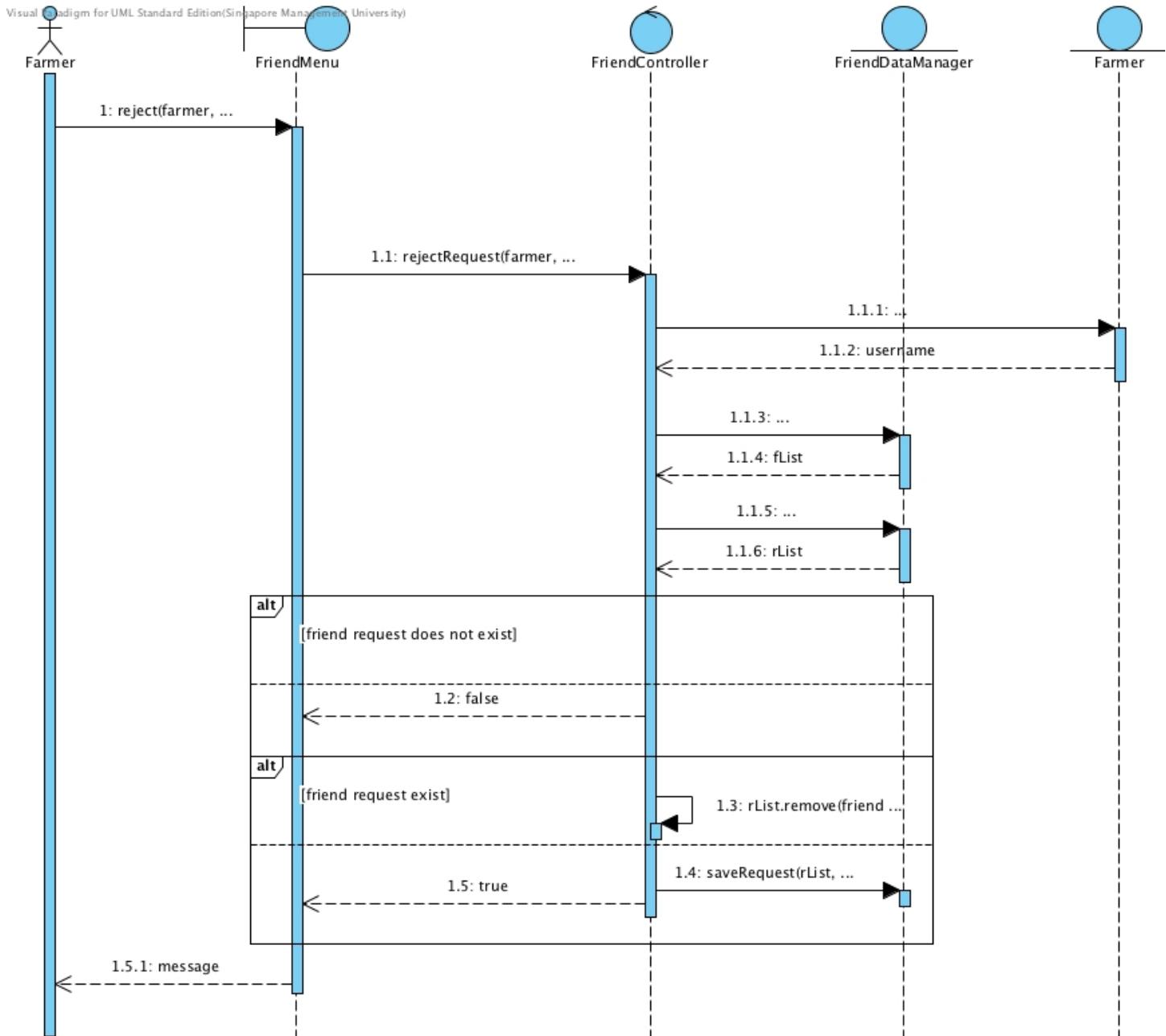
- 2a. The friend request does not exist
 - ix. The system informs farmer that his/her friend request does not exist
 - x. The use case ends.

1.4.5.2 System Sequence Diagram



PROJECT - Farm City

1.4.5.3 Sequence Diagram



PROJECT - Farm City

1.4.6 Unfriend

1.4.6.1 Use Case Scenario

Actor: Farmer

Precondition: The farmer is logged in.

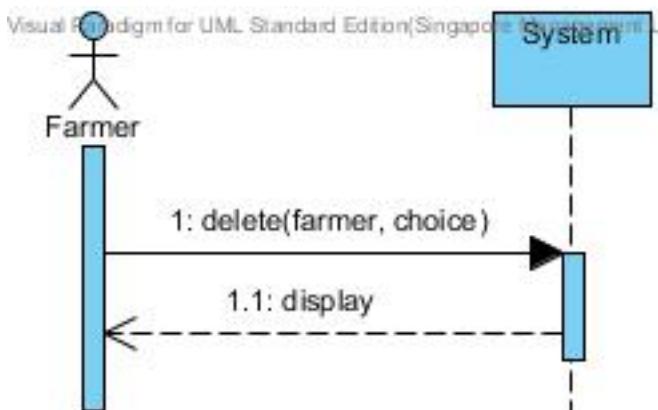
Main flow of events:

27. The use case begins when farmer chooses to delete a friend
28. The system projects the user to enter who he/she wants to unfriend
29. The system deletes the friend from the farmer's friend list and the farmer from the friend's friend list
30. Use case ends.

Alternate flows:

- 2a. The friend does not exist in the farmer's friend list
 - xi. The system informs farmer that the friend does not exist in the farmer's friend list
 - xii. The use case ends.

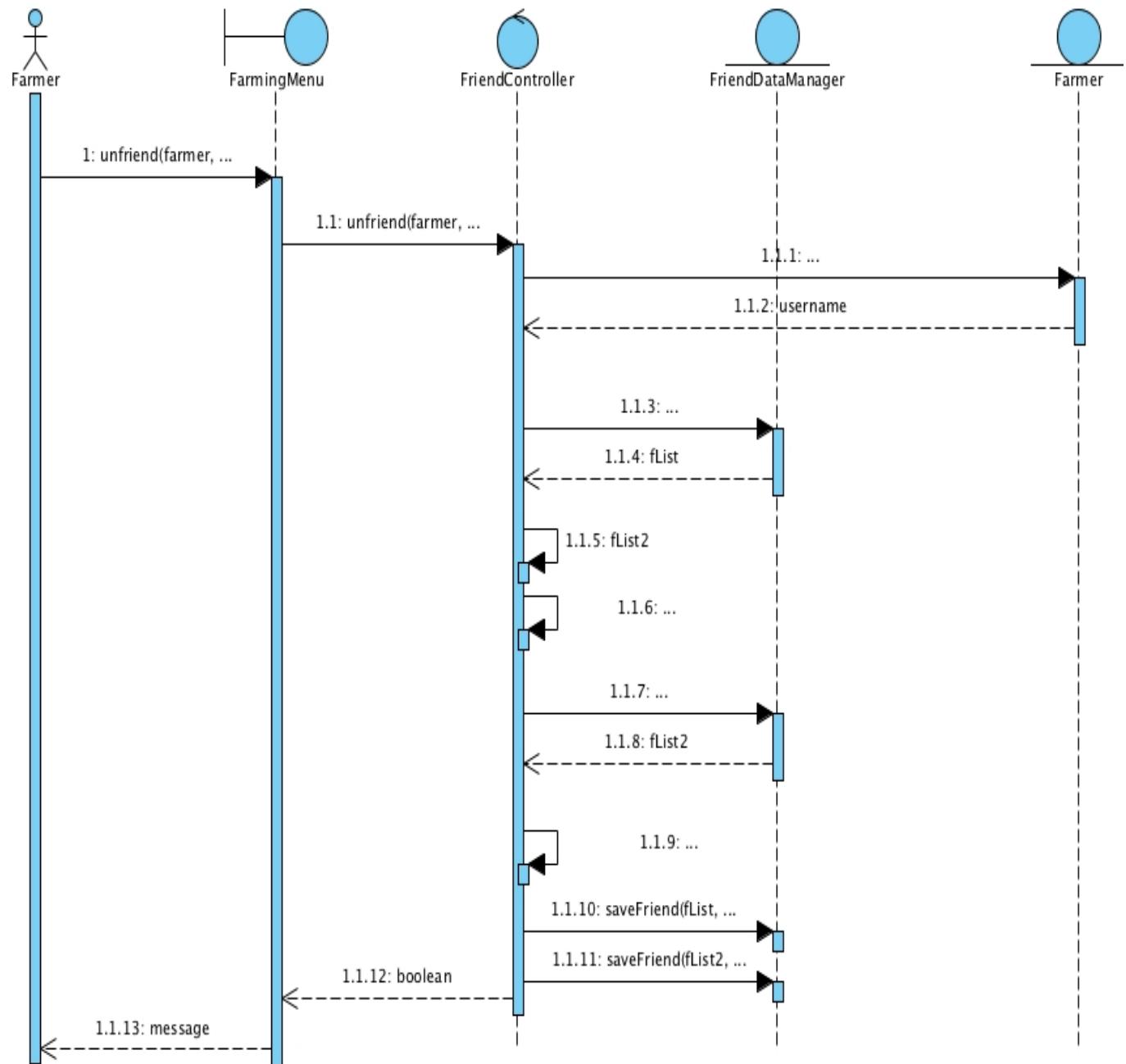
1.4.6.2 System Sequence Diagram



1.4.6.3 Sequence Diagram

PROJECT - Farm City

UML Paradigm for UML Standard Edition(Singapore Management University)



PROJECT - Farm City

1.4.7 Plant

1.4.7.1 *Use Case Scenario*

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

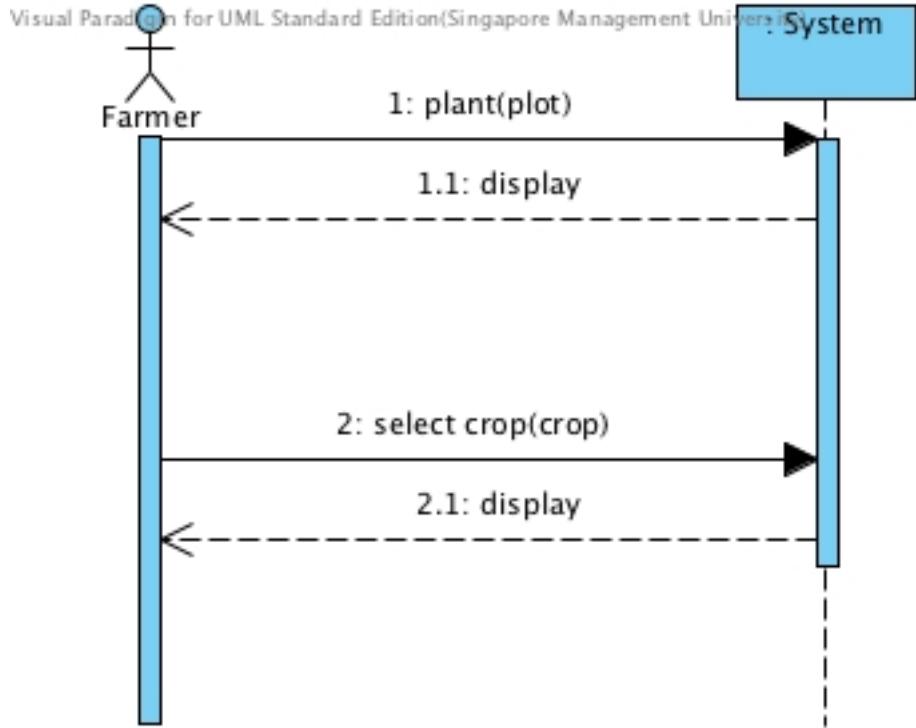
31. The use case begins when farmer chooses to plant and selects a plot.
32. The system displays the seeds from farmer's inventory.
33. The farmer chooses a crop to plant.
34. The system displays the information of the planting and records the plant date.
35. Use case ends.

Alternate flows:

- 2a. The plot farmer selected is not empty.
 - xiii. The system informs farmer that the plot is not available.
 - xiv. Use case ends.
- 3a. The farmer has an empty inventory.
 - ix. The system informs farmer that there is no seed can be planted.
 - x. Use case ends.

PROJECT - Farm City

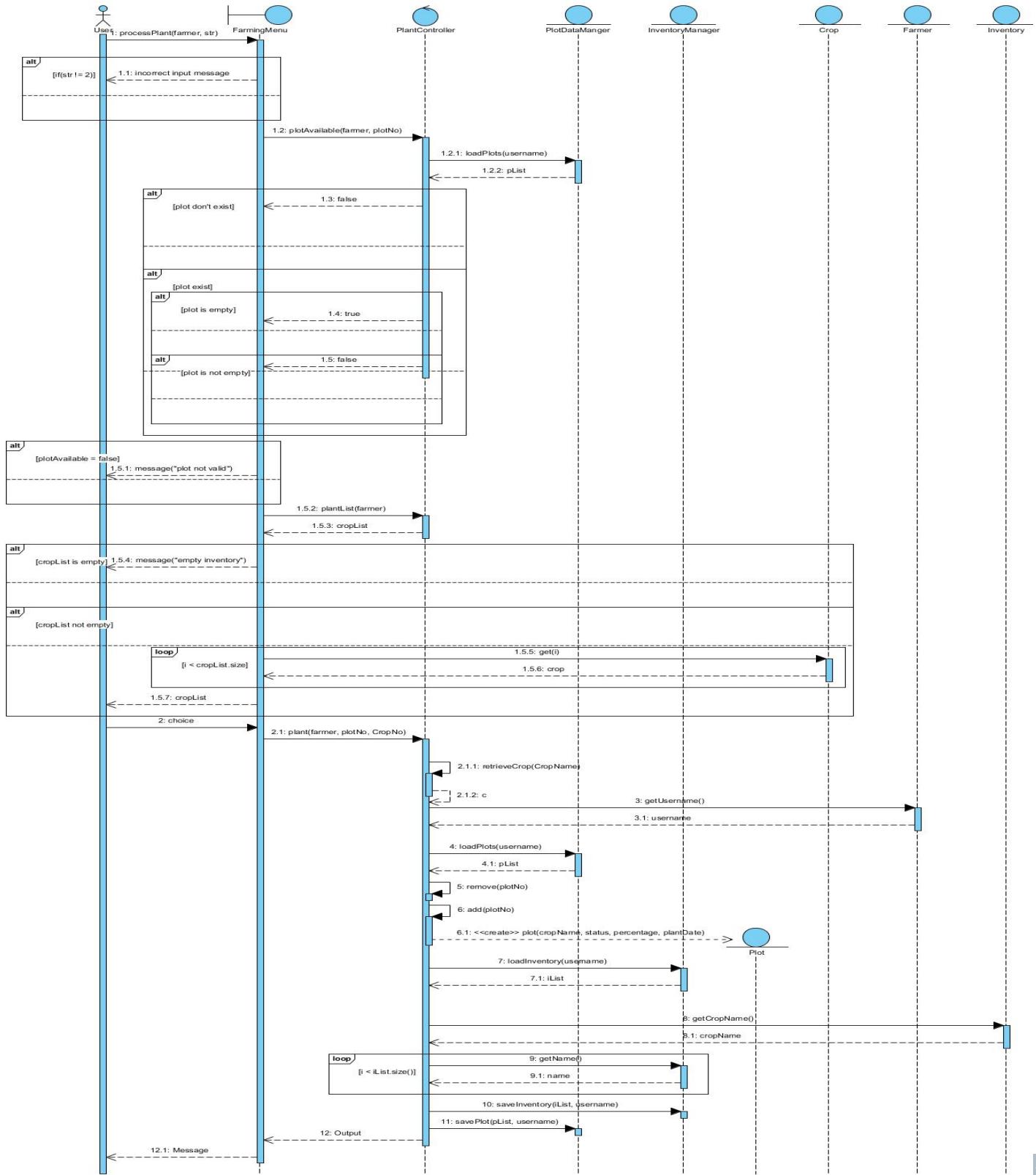
1.4.7.2 System Sequence Diagram



1.4.7.3 Sequence Diagram

PROJECT - Farm City

UML Paradigm for UML Standard Edition (Singapore Management University)



PROJECT - Farm City

1.4.8 Harvest

1.4.8.1 Use Case Scenario

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

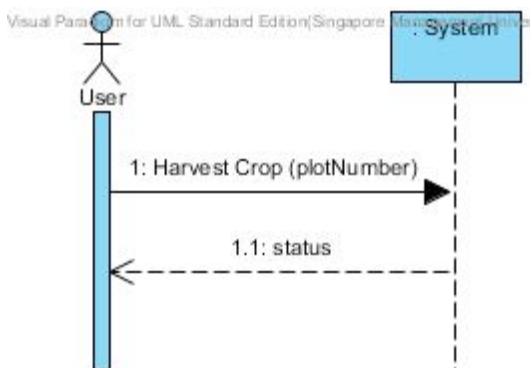
1. The use case begins when farmer chooses to harvest in the My Farm page.
2. The system harvests crops that are ready for harvest on any plot of land.
3. The system displays the amount of harvest and the XP and gold the farmer earns.
4. The system adds the XP and gold to the farmer's account.
5. Use case ends.

Alternate flow:

- 2a. Farmer does not have any crop ready for harvest.

- i. The system informs the farmer that no crop can be harvested.
- ii. Use case ends.

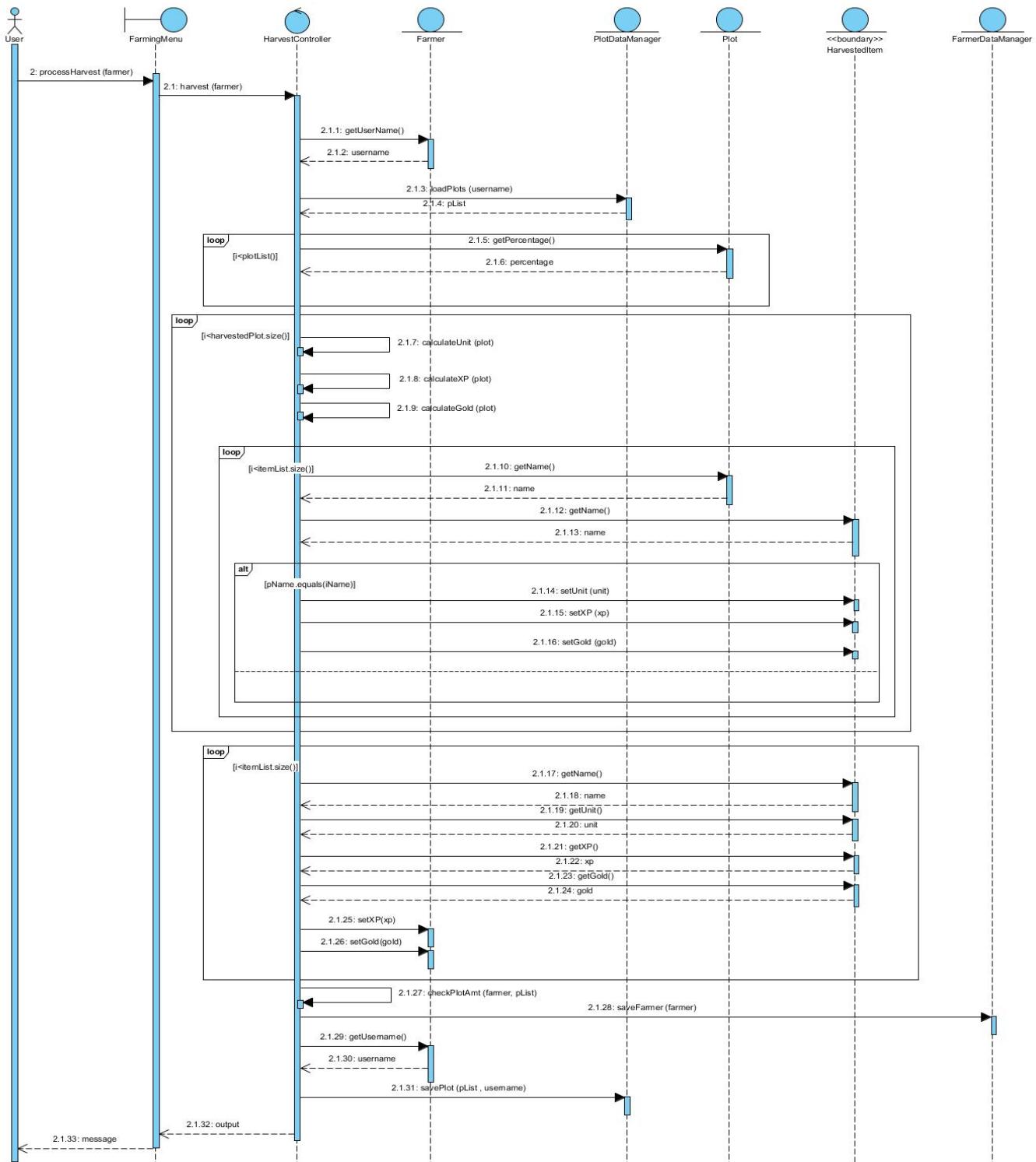
1.4.8.2 System Sequence Diagram



PROJECT - Farm City

1.4.8.3 Sequence Diagram

UML Paradigm for UML, Standard Edition(Singapore Management University)



PROJECT - Farm City

1.4.9 Clear

1.4.9.1 *Use Case Scenario*

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

1. The use case begins when farmer choose to clear a wilted crop
2. The system check if the plot number is valid
3. The system check if the farmer have enough gold to clear a crop
4. The system clear the selected plot number wilted crop
5. The system deduct 5 gold from the farmer
6. The system inform the user that the selected wilted crop has been cleared and 5 gold has been deducted from his/her balance
7. Use case ends.

Alternate flows:

2a. The plot number entered does not exist

xv. The system informs farmer that the plot number does not exist

xvi. The use case ends.

3a. The farmer does not have enough gold in his/her balance to clear crop

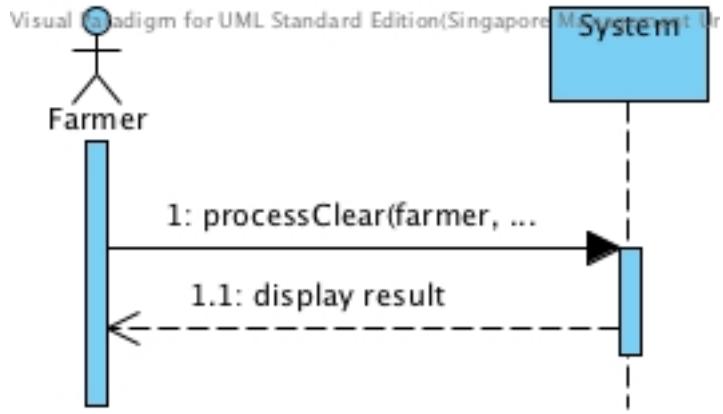
xi. The system clear all the farmer's wilted crop

xii. The system inform the farmer that he/she does not have enough gold to clear his/her selected crop and a super power has been activated to help him/her clear all his wilted plots for free

xiii. The use case ends

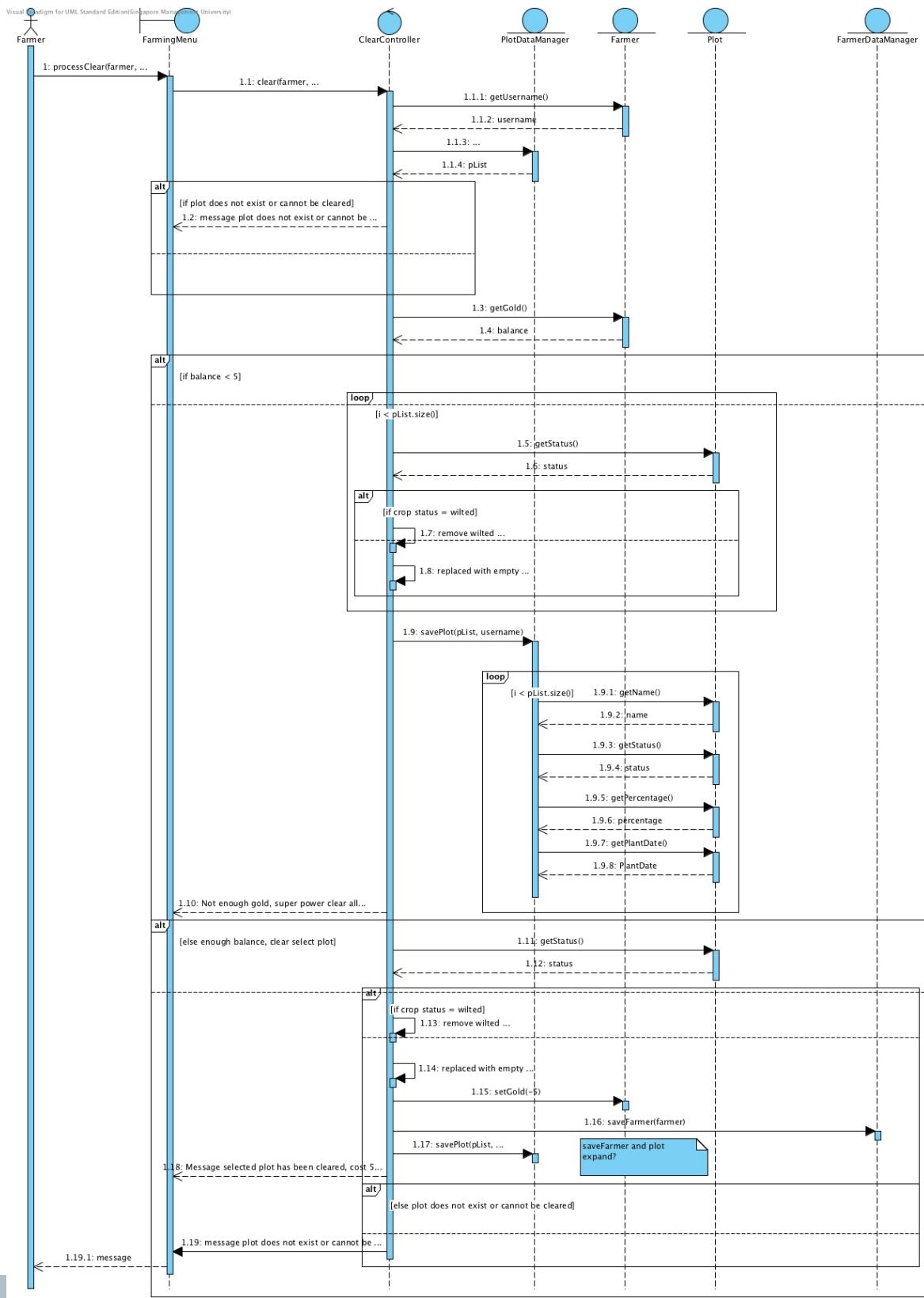
PROJECT - Farm City

1.4.9.2 System Sequence Diagram



1.4.9.3 Sequence Diagram

PROJECT - Farm City



PROJECT - Farm City

1.4.10 Buy Seeds

1.4.10.1 Use Case Scenario

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

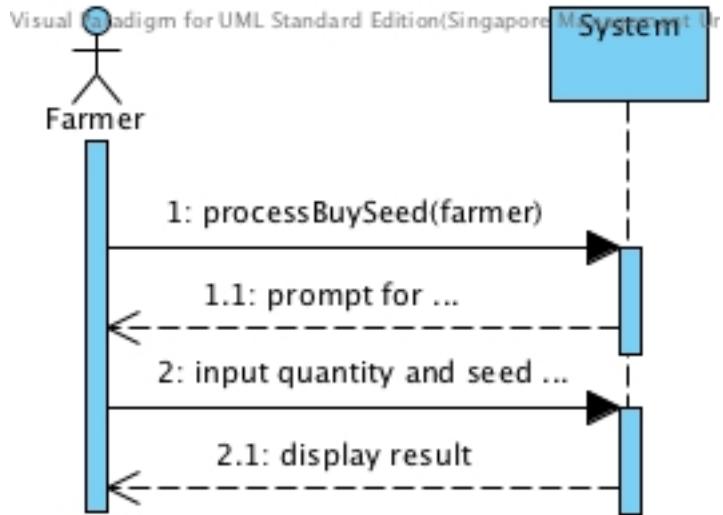
36. The use case begins when farmer chooses buy seed(s)
37. The system print out all seeds available for sale
38. The farmer enter his/her choice of seed and quantity
39. The system deduct the total cost for the seed and the given quantity
40. The system add the seed into the farmer's inventory
41. Use case ends.

Alternate flows:

- 3a. The choice/quantity entered is invalid
 - xvii. The system informs farmer that his/her choice/quantity is invalid
 - xviii. The use case ends.
- 4a. The farmer does not have enough gold in his/her balance to buy seeds
 - xiv. The system informs farmer that his/her balance is not enough to buy seeds
 - xv. The use case ends

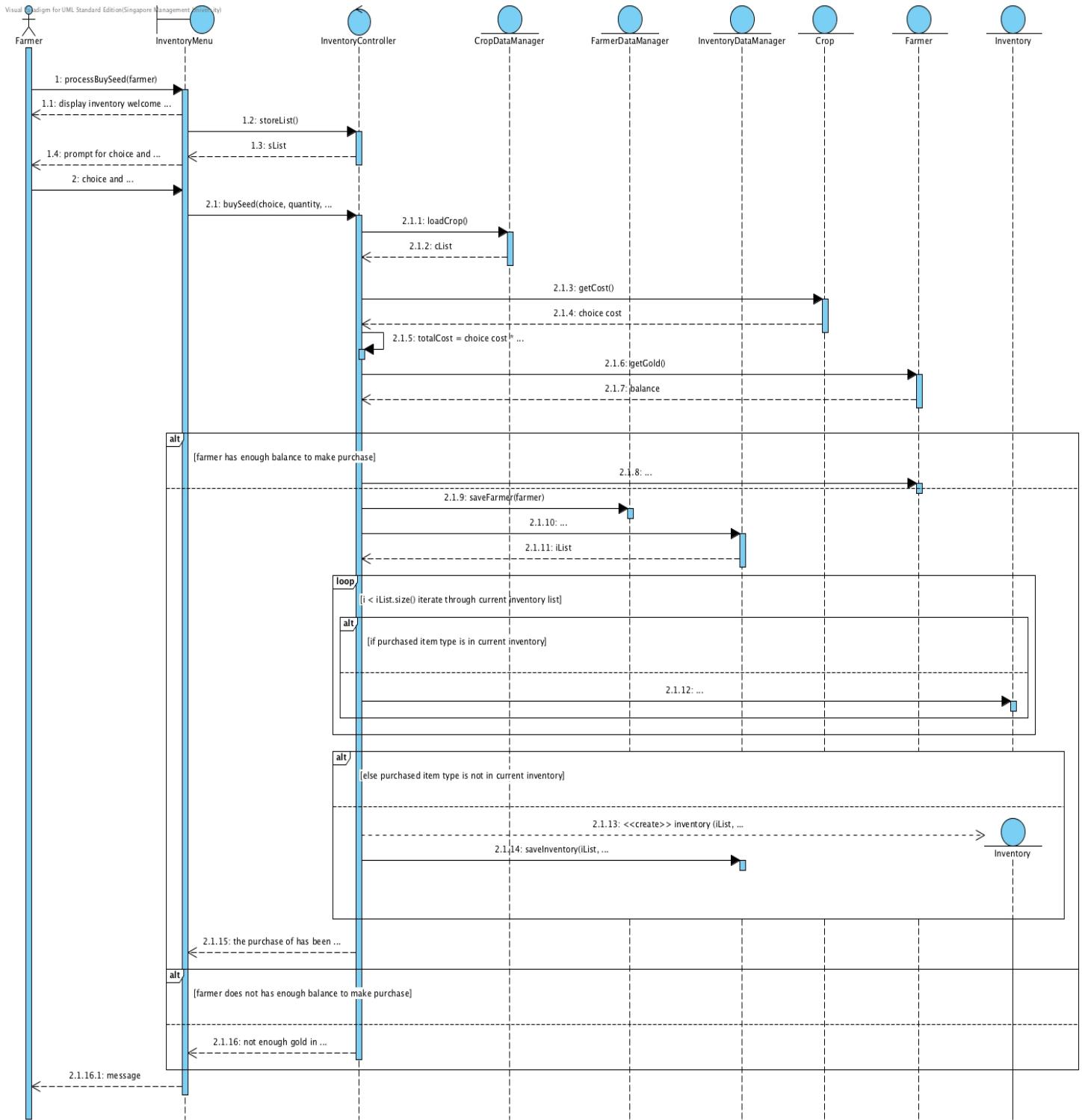
PROJECT - Farm City

1.4.10.2 System Sequence Diagram



1.4.10.3 Sequence Diagram

PROJECT - Farm City



PROJECT - Farm City

1.4.11 Send Gift

1.4.11.1 Use Case Scenario

Actor: Farmer

Precondition: The farmer is logged in.

Main flow of events:

8. The use case begins when farmer chooses to gift seeds to friends.
9. The system displays the seeds available for gifting.
10. The farmer chooses a seed to gift.
11. The farmer enter one or more recipient name's, defining "," between usernames for more than one recipient.
12. The system displays the message if gifting is successful.
13. Use case ends.

Alternate flows:

4a. The user has no seed in his inventory

- i. The system informs the user, that user inventory is empty
- ii. The system brings the user back to the inventory page
- iii. Use case ends

4b. The user enters user's username, to send gift to own self.

- i. The system informs the user, that use cannot send gift to yourself
- ii. The system brings the user back to the inventory page
- iii. Use case ends

4c. The user enters a username not found in his friend list / user has no friends

- i. The system informs the user, specifically which friend is not in his friend list
- ii. The system brings the user back to the inventory page
- iii. Use case ends

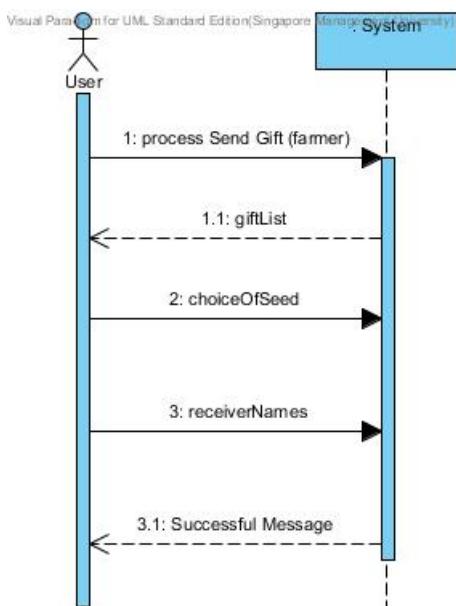
4d. The user tries to send a recipient twice in a day

- i. The system informs the user, that user has already sent a gift to recipient for the day

PROJECT - Farm City

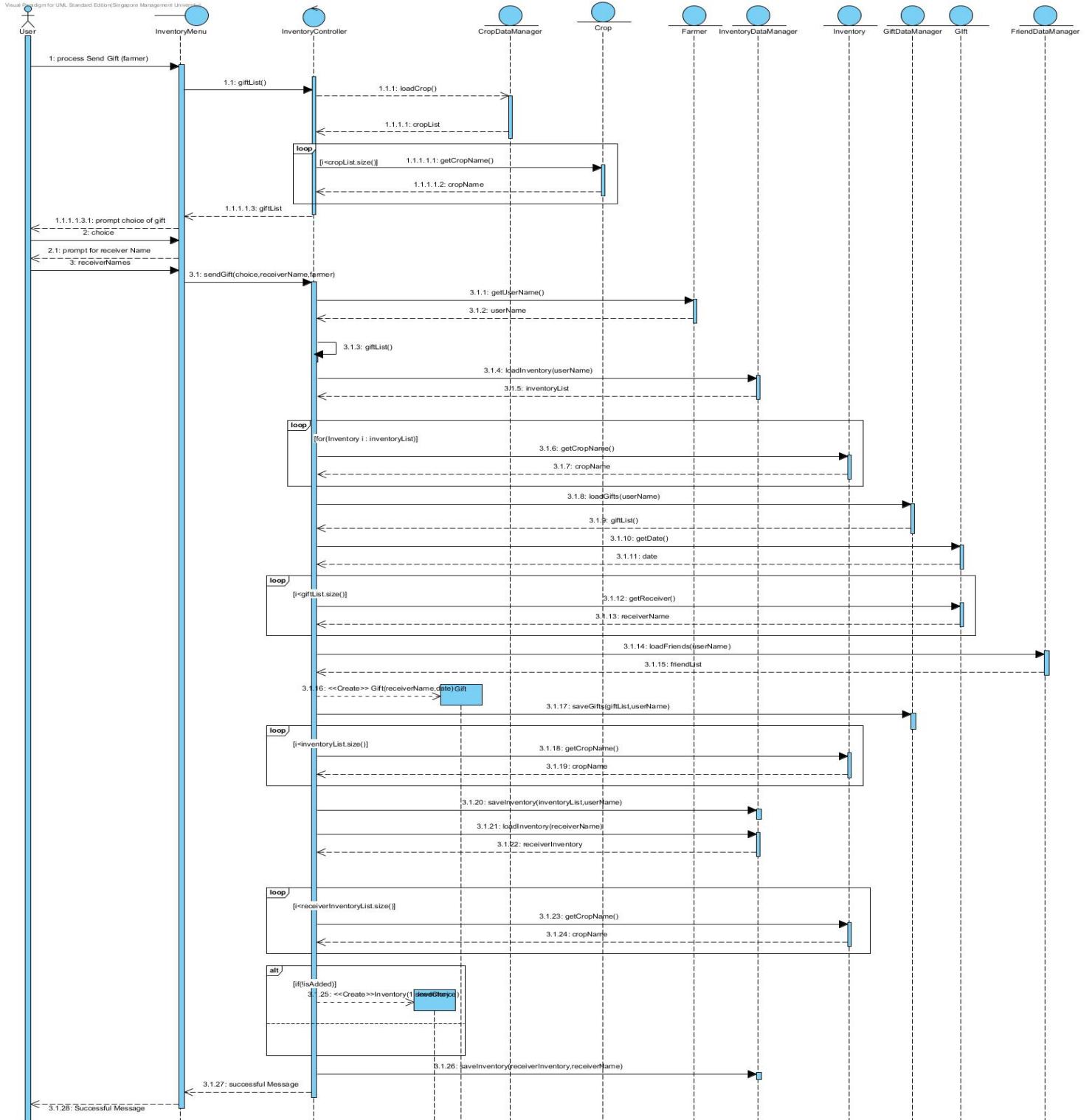
- ii. The system brings the user back to the inventory page
 - iii. Use case ends
- 4e. The amount of recipient is more than the user choice of seeds
- iv. Following the user's order of recipients, the system only gift to the first few
 - v. The system informs the user, the success of gifting for each friend
 - vi. The system brings the user back to the inventory page
 - vii. Use case ends
- 4f. The user sends/has send more than 5 gifts in a day
- i. The system sends gifts up to 5 gifts a day, once the user reaches maximum for the day, user cannot send gifts
 - ii. The system informs the user, that the user has reached the maximum for the day
 - iii. The system brings the user back to the inventory page
 - iv. Use case ends

1.4.11.2 System Sequence Diagram



PROJECT - Farm City

1.4.11.3 Sequence Diagram



PROJECT - Farm City

1.5 Object Oriented Design Consideration

During the whole process of developing the application, we always followed the Single Responsibility Principle (SRP) to design. The classes were divided into several different types such like entity, controller, data manager, menu, exception and application. For each class, there is only one function. For example, the FriendDataManager.java is only able to handle the data related to “Friend”, but doesn’t have any access to the database related to other sectors in the game such like “Farming”. The Farmer.java only represents the registered user with basic information of his/her account, but cannot reflect the situation of farmer’s plots, which should be represented by the Plot.java.

We chose to develop the application in the Iterative Process instead of the Waterfall Process. As a beginner to OOAD, though it’s a small project, we still felt that it wasn’t very easy. So we decided to choose Iterative Process so that we can have more chances between iterations to think about our design and make a better plan. After the project being released, we estimated the workload of each section and divided the whole project into three parts. In the Iteration 1 we finished the “My Farm” part with “plant”, “harvest” and “clear”, and in the Iteration 2 we finished the “My Friend” part with “request”, “accept”, “reject” and “unfriend”. In the Iteration, we finished the “My Inventory” part with “buy seeds” and “send gifts”. We also used the time boxing to make sure that our project is carried out efficiently, which was really helpful for the project management. We met together to nail down the design based on the requirement for each iteration, and split the work.

PROJECT - Farm City

2 MISCELLANEOUS

2.1 The Challenge

One big challenge we faced is that we were lacking in experience. As the beginners who worked on a project for the first time, it took us a lot of time to search information and get the project started. For example, to record the time when the crop is planted and to calculate the percentage based on the time, it took us a great effort to find the `SimpleDateFormat`.

Another challenge we faced is compilation of our codes. As we split the project into 3 section – “My Friends”, “My Farm”, and “My Inventory”, each of us coded the boundary, controller and entity classes for our individual parts. There are times, when a controller needs to retrieve data from other data managers, the methods does not produce the expected output. This has led to multiple meetings for clarification of methods and countless occurrence of modification of codes, and this causes a wave of code modification, which was the “most challenging” issue faced in a collaborative coding project.

2.2 What we have learnt

Collaboration: we need the collaboration to get rid of all the difficulties and finished the project together.

Teamwork: we have learnt how to work with each other in the project team.

Communication: When we have different opinion, we have to know how to communicate efficiently and effectively.

Knowledge application: the knowledge from OOAD class is actually not very hard to understand. But it's not east to apply the knowledge and use it in our project. After the project, we all know the importance of knowledge application and gained some experience from the project.

PROJECT - Farm City

3 SCREENSHOTS OF SAMPLE RUNS

3.1 Register

Main flow

1. The User enters “1” to select <Register> on the Welcome Page.

The System display the Farm City Registration Page and prompts the User to enter his/her desired username

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

-- Farm City :: Registration --
Enter your username >
```

2. The User enters a his/her desired username “JavaProject”

```
-- Farm City :: Registration --
Enter your username > JavaProject
Enter your full name >
```

The System accepts the user’s input and prompts the User to enter his/her “full name”

3. The User enters his full name “John Smith Jian Yong”

```
Enter your full name > John Smith Jian Yong
Enter your password >
```

The System accepts the user’s input and prompts the User to enter a password

4. The User enters his desired password “ilovejava”

```
Enter your password > ilovejava
Confirm your password >
```

The System accepts the User’s input and prompts the User to confirm his password

PROJECT - Farm City

5. The User enters confirmed his password as "ilovejava"

The System confirms the password and prompts the User that his account has been successful created

The System displays the Welcome Page again and prompts the User to enter his/her next choice

```
Confirm your password > ilovejava
Hi, JavaProject! Your account is successfully created!

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

Alternate flows:

- 2a. The User enter a username that is not alphanumeric e.g. "!notAlphaNumeric7"

The System informs the user that the username entered is invalid and username should contain only alphanumeric characters

```
-- Farm City :: Registration --
Enter your username > !notAlphaNumeric7
Invalid username! Username should contain only alphanumeric characters.
-- Farm City :: Registration --
Enter your username >
```

The System prompts the User to enter a desired username again

- 2b. The User entered a username that is already in the System

The System informs the User that the username entered already exist and tells that User that username is required to be unique

```
-- Farm City :: Registration --
Enter your username > ilovejava
Username already exists! Username is required to be unique.

-- Farm City :: Registration --
Enter your username >
```

The System prompts the User to enter a desired username again

PROJECT - Farm City

- 4a. The User entered a comfirm password is different the first password

The System informs the User that the passwords did not match

```
Enter your password > a
Confirm your password > b
Password does not match! Please confirm your password again.
Confirm your password >
```

The System prompts the User to confirm his/her password again

3.2 Login

Main flow

1. The User enters “2” to select <Login> on the Welcome Page.

The System prompts the User to enter his/her username

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

Enter your username >
```

2. Ther Use enters “JavaProject” as his username

The System accepts the User’s input and prompts him to enter the password

```
Enter your username > JavaProject
Enter your password >
```

3. The Use enters a correct password “ilovejava” which matches with the username “JavaProject”

```
Enter your password > ilovejava

-- Farm City :: Main Menu --
Welcome, John Smith Jian Yong!

1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice >
```

The System Login to the User account and displays his Main Menu

PROJECT - Farm City

Alternate flows:

- 2a. The User enter a username that does not exist in the System e.g. “!monster”
The System prompts the user to enter a password

```
Enter your username > monster
Enter your password > a
The username you enter does not exist. Please try again.
Enter your username >
```

The System informs the User that the username does not exist and ask him or her to type again
The System prompts the User to enter a username again

- 2b. The User entered a password that does not match with the username (username “JavaProject” password does not match)
The System informs the User that the Password is Wrong and ask him/her to try again

```
Enter your username > JavaProject
Enter your password > a
Wrong Password! Please try again.
Enter your username >
```

The System prompts the User to enter a username again

3.3 Send Friend Request

Main flow

Prerequisite: The Farmer is already logged in

1. The Farmer enters “1” to select <My Friends> on the Main Menu
The System displays the Friend Menu and prompts the Farmer to make a choice
The Farmer enters “q” to select <re[Q]uest> on the Friend Menu
The System prompts the Farmer to enter the username of the Farmer he/she wants to be friends with

PROJECT - Farm City

```
-- Farm City :: Main Menu --
Welcome, John Smith Jian Yong!

1. My Friends
2. My Farm
3. My Inventory
4. Logout
Enter your choice > 1

-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
You have no friend request.

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

Enter the username >
```

2. The System prompts the Farmer to enter his/her username

The Farmer enters a username “javaclass”

The System accepts the input and informs the user that a friend request has been sent to the username entered

The System shows the My Friends menu to the Farmer

```
Enter the username > javaclass
A friend request is sent to javaclass.

-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
You have no friend request.

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

PROJECT - Farm City

In the other Farmer My Friends Menu, it shows the request he/she received under My Requests

```
-- Farm City :: My Friends --
Welcome, Janice

My Friends:
You have no friend.

My Requests:
1. javaproject

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

Alternate flows:

- 2a. The Farmer is trying to send a friend request to a username that he/she had sent before (e.g. Farmer javaclass)

The System informs the Farmer that he/she have sent the same request before

The System shows the My Friends menu to the Farmer

```
Enter the username > javaclass
You have sent the same request before.

-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
You have no friend request.

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

- 2b. The Farmer is trying to send a friend request to a username that does not exist (e.g. Farmer dancingqueen)

The System informs the Farmer that the username doesn't exist and prompts the Famer to try again

PROJECT - Farm City

The System shows the My Friends menu to the Farmer

```
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > Q
Enter the username > thewrongname
Username doesn't exist. Please try again.

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

My Friends:
1. kenneth

My Requests:
You have no friend request.

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

- 2c. The Farmer is trying to send a friend request to a username that does not exist (e.g. Farmer dancingqueen)
- The System informs the Farmer that the username doesn't exist and prompts the Famer to try again
- The System shows the My Friends menu to the Farmer

```
Enter the username > javaproject
You cannot send request to yourself. Please try again.

== Farm City :: My Friends ==
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
You have no friend request.

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

- 2d. The Farmer is trying to send a friend request to a username that he/she is already friend with (e.g. javaaclass)
- The System informs the Farmer that the username doesn't exist and prompts the Famer to try again
- The System shows the My Friends menu to the Farmer

PROJECT - Farm City

```
Enter the username > javaclass
This farmer has already been your friend.

-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
1. javaclass

My Requests:
You have no friend request.

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

3.4 Accept Friend Request

Main flow

Prerequisite: The Farmer is already logged in

1. The Farmer enters “1” to select <My Friends> on the Main Menu
The System displays the Friend Menu and prompts the Farmer to make a choice
The Farmer enters “A1” to select <[A]ccept> on the Friend Menu > where 1 is the number associated with his/her requestor by the System in the Request List
The System informs the Farmer that the selected requestor is now his/friend and the requestor has been added to the Farmer’s Friend List

PROJECT - Farm City

```
-- Farm City :: My Friends --
Welcome, Janice

My Friends:
You have no friend.

My Requests:
1. javaproject

[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > A1

javaproject is now your friend.

-- Farm City :: My Friends --
Welcome, Janice

My Friends:
1. javaproject

My Requests:
You have no friend request.

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

The Requestor is also friends with the Farmer in his/her Friend List.

```
-- Farm City :: My Friends --
Welcome, Janice

My Friends:
1. javaproject

My Requests:
You have no friend request.

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

Alternate flows:

- 2a. The Farmer is trying to be accept a friend request that does not exist
The System informs the Farmer that the friend cannot be found

PROJECT - Farm City

The System shows the My Friends menu to the Farmer

```
-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
1. javaclass

My Requests:
You have no friend request.

[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > A2

The farmer does not exist.
```

3.5 Reject Friend Request

Main flow

Prerequisite: The Farmer is already logged in

1. The Farmer enters “1” to select <My Friends> on the Main Menu

The System displays the Friend Menu and prompts the Farmer to make a choice

The Farmer enters “R1” to select <[R]eject> on the Friend Menu > where 1 is the number associated with his/her requestor by the System in the Request List

```
-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
1. javaclass

[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > R1
```

2. The System removes the selected request from the Farmer’s Request List and informs the Farmer that the friend request has been rejected successfully

```
-- Farm City :: My Friends --
Welcome, John Smith Jian Yong

My Friends:
You have no friend.

My Requests:
You have no friend request.

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

Alternate flows:

PROJECT - Farm City

- 2a. The Farmer is trying to be accept a friend request that does not exist

The System informs the Farmer that the request does not exist

The System shows the My Friends menu to the Farmer

```
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > R1
```

```
The request you choose doesn't exist.
```

```
-- Farm City :: My Friends --
Welcome, John Smith Jian Yong
```

```
My Friends:
You have no friend.
```

```
My Requests:
You have no friend request.
```

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

3.6 Unfriend

Main flow

Prerequisite: The Farmer is already logged in

1. The Farmer enters “1” to select <My Friends> on the Main Menu

The System displays the Friend Menu and prompts the Farmer to make a choice

The Farmer enters “U1” to select <[U]nfriend> where 1 is the number associated with his/her friend by the System in the Friend List

```
-- Farm City :: My Friends --
Welcome, John Smith Jian Yong
```

```
My Friends:
1. javaaclass
```

```
My Requests:
You have no friend request.
```

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

2. The System informs the Farmer that the unfriend has been successfully and returns him/her to the My Friends Menu

PROJECT - Farm City

```
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > U1  
Unfriend successfully!  
== Farm City :: My Friends ==  
Welcome, John Smith Jian Yong  
  
My Friends:  
You have no friend.  
  
My Requests:  
You have no friend request.  
  
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject >
```

In the unfriended Farmer's Friend List (javaclass) the unfriender (javaproject) is also remove from his/her Friend List

```
-- Farm City :: My Friends --  
Welcome, Janice  
  
My Friends:  
You have no friend.  
  
My Requests:  
You have no friend request.  
  
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject >
```

Alternate flows:

- 2a. The Farmer is trying to unfriend a Farmer that doesn't exist in his/her friend list (e.g. javaclass)
The System informs the Farmer that the friend cannot be found
The System shows the My Friends menu to the Farmer

```
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject > U1  
The friend cannot be found.  
  
== Farm City :: My Friends ==  
Welcome, John Smith Jian Yong  
  
My Friends:  
You have no friend.  
  
My Requests:  
You have no friend request.  
  
[M]ain | [U]nfriend | re[Q]uest | [A]ccept | [R]eject >
```

PROJECT - Farm City

3.7 Plant

Main flow

Prerequisite: The Farmer is already logged in

1. The Farmer enters “2” to select <My Farm> on the Main Menu

The System displays the Farm Menu and prompts the Farmer to make a choice

The Farmer enters “P1” to select plot 1 and the option <[P]lant> on the Farm Menu where 1 is the number associated with the Farmer’s plot number

```
-- Farm City :: My Farm --
Welcome, John Smith Jian Yong!
Rank: Novice           Gold: 980

You have 5 plots of land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > P1
```

2. The System shows the Farmer’s crop list and prompts the Farmer to select one

```
Select the crop:
1. Papaya
[M]ain | Select Choice >
```

3. The Farmer enters “1” to select the seed he wants to plant (papaya)

The System remove the seed from the Farmer’s inventory and plant the seed on the select plot (plot 1)

The System bring the Farmer back to the Farm Menu

```
-- Farm City :: My Farm --
Welcome, John Smith Jian Yong!
Rank: Novice           Gold: 980

You have 5 plots of land.
1. Papaya      [-----]    0%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest >
```

Alternate flows:

PROJECT - Farm City

- 1a. The Farmer is trying to plant a seed with an empty inventory

The System informs the Farmer that his/her inventory is empty and informs the Farmer that it will return to the Farm Menu

```
You have 5 plots of land.  
1. Papaya      [###-----]    35%  
2. <empty>  
3. <empty>  
4. <empty>  
5. <empty>  
[M]ain | [P]lant | C[L]ear | [H]arvest > P2  
  
Your inventory is empty. Please buy seeds from store.  
You will return to the Farm Page.
```

- 1b. The Farmer is trying to plant a seed on a planted plot or a plot that does not exist

The System informs the Farmer that the plot he/she has chosen is not empty or does not exist and return the Farmer to the Farm Menu

(On a planted plot)

```
-- Farm City :: My Farm --  
Welcome, John Smith Jian Yong!  
Rank: Novice          Gold: 980  
  
You have 5 plots of land.  
1. Papaya      [###-----]    35%  
2. <empty>  
3. <empty>  
4. <empty>  
5. <empty>  
[M]ain | [P]lant | C[L]ear | [H]arvest > P1  
  
The plot you choose is not empty or does no exist.  
You will return to the Farm Page.
```

(On a plot that does not exist) e.g. P6 when the last plot is P5

PROJECT - Farm City

```
-- Farm City :: My Farm --
Welcome, John Smith Jian Yong!
Rank: Novice          Gold: 980

You have 5 plots of land.
1. Papaya      [#####-----]    43%
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > P6

The plot you choose is not empty or does no exist.
You will return to the Farm Page.
```

3.8 Harvest

Main flow

Prerequisite: The Farmer is already logged in

1. From "My Farm" menu, the player enters H to harvest plots that are 100.

PROJECT - Farm City

```
Enter your choice > 2

== Farm City :: My Farm ==
Welcome, Apple Ng!
Rank: Novice                                     Gold: 530

You have 5 plots of land.
1. Papaya      [#####----]    100%
2. Pumpkin     [#####---]    54%
3. Watermelon  [-----]    13%
4. Watermelon  [-----]    13%
5. Sunflower   [##-----]    27%
[M]ain : [P]lant : C[L]ear : [H]arvest > H

You have harvested 95 units of Papaya for 8 XP and 15 gold.

== Farm City :: My Farm ==
Welcome, Apple Ng!
Rank: Apprentice                                Gold: 545

You have 6 plots of land.
1. <empty>
2. Pumpkin     [#####---]    55%
3. Watermelon  [-----]    13%
4. Watermelon  [-----]    13%
5. Sunflower   [##-----]    27%
6. <empty>
[M]ain : [P]lant : C[L]ear : [H]arvest >
```

2. The System harvest the plots, adding the gained experience and gold corresponding to harvested plots to the user.
3. The system level up and add plots to the user if the user has sufficient experience.
4. The system resets the harvested plot to empty.
5. The system displays the message of the experience and gold gained from harvest.

Alternate flows

- 2a. When there is no plot that can be harvested, the System displays the information.

PROJECT - Farm City

```
-- Farm City :: My Farm --
Welcome, Test!
Rank: Novice                      Gold: 1000

You have 5 plots of land.
1. <empty>
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > H

No plot can be harvested yet.

-- Farm City :: My Farm --
Welcome, Test!
Rank: Novice                      Gold: 1000

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

3.9 Clear

Main flow

Prerequisite: The Farmer is already logged in

1. From “My Farm” menu, the player enters L<ID> to go the “Clear” menu.

```
-- Farm City :: My Farm --
Welcome, Kenneth LEE!
Rank: Grandmaster                  Gold: 910

You have 8 plots of land.
1. Sunflower      [ wilted   ]
2. Pumpkin        [ wilted   ]
3. <empty>
4. <empty>
5. <empty>
6. <empty>
7. <empty>
8. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > L1
```

PROJECT - Farm City

2. The System displays the information to confirm the plot has been cleared and displays the cost.

```
The plot you picked has been cleared.  
This operation costed you 5 gold coins.  
  
== Farm City :: My Farm ==  
Welcome, Kenneth LEE!  
Rank: Grandmaster Gold: 905  
  
You have 8 plots of land.  
1. <empty>  
2. Pumpkin [ wilted ]  
3. <empty>  
4. <empty>  
5. <empty>  
6. <empty>  
7. <empty>  
8. <empty>  
[M]ain | [P]lant | C[L]ear | [H]arvest >
```

3. Then the System will be back to “My Farm” menu.

Alternate flows

- 2a. When the player chooses a plot that is empty, the System displays the information that the plot cannot be cleared.

```
== Farm City :: My Farm ==  
Welcome, Kenneth LEE!  
Rank: Grandmaster Gold: 905  
  
You have 8 plots of land.  
1. <empty>  
2. Pumpkin [ wilted ]  
3. <empty>  
4. <empty>  
5. <empty>  
6. <empty>  
7. <empty>  
8. <empty>  
[M]ain | [P]lant | C[L]ear | [H]arvest > L3  
  
The plot you picked cannot be cleared yet.
```

- 2b. When the player chooses a plot un-wilted crop, the System displays the information that the plot cannot be cleared yet.

PROJECT - Farm City

```
== Farm City :: My Farm ==
Welcome, Kenneth LEE!
Rank: Grandmaster                      Gold: 905

You have 8 plots of land.
1. Pumpkin      [-----]    2%
2. Pumpkin      [ wilted ] 
3. <empty>
4. <empty>
5. <empty>
6. <empty>
7. <empty>
8. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > L1

The plot you picked cannot be cleared yet.
```

- 2c. When the player has zero gold and chooses to clear a plot, the System will clear all plots with wilted crop by a “Magic Power”.

```
== Farm City :: My Farm ==
Welcome, TEST1!
Rank: Novice                           Gold: 0

You have 5 plots of land.
1. Pumpkin      [ wilted ] 
2. <empty>
3. <empty>
4. <empty>
5. <empty>
[M]ain | [P]lant | C[L]ear | [H]arvest > L1

Your gold is not enough.
A super power helps you clear all wilted plots free of cost.
```

3.10 Buy Seeds

Main flow

Prerequisite: The Farmer is already logged in

1. From “My Inventory” menu, the player enters B to go to “Buy Seeds” menu. The System displays the seeds available in store.

PROJECT - Farm City

```
== Farm City :: My Inventory ==
Welcome, TEST3!
Rank: Novice           Gold: 1000

My Seeds:
You have no seed in your inventory.

[M]ain | [B]uy | [G]ift | Select choice > B
== Farm City :: My Inventory ==
Welcome, TEST3!
Rank: Novice           Gold: 1000

Seeds Available:
1. Papaya costs: 20 gold
   Harvests in: 30 mins
   XP Gained: 8
2. Pumpkin costs: 30 gold
   Harvests in: 60 mins
   XP Gained: 5
3. Sunflower costs: 40 gold
   Harvests in: 120 mins
   XP Gained: 20
4. Watermelon costs: 50 gold
   Harvests in: 240 mins
   XP Gained: 1

[M]ain | Select choice >
```

2. The player enters the choice and quantity of seeds. The System displays the information of the purchase and returns to “My Inventory” menu to display the player’s inventory.

```
Seeds Available:
1. Papaya costs: 20 gold
   Harvests in: 30 mins
   XP Gained: 8
2. Pumpkin costs: 30 gold
   Harvests in: 60 mins
   XP Gained: 5
3. Sunflower costs: 40 gold
   Harvests in: 120 mins
   XP Gained: 20
4. Watermelon costs: 50 gold
   Harvests in: 240 mins
   XP Gained: 1

[M]ain | Select choice > 1
Enter quantity > 2
2 bags of seeds purchased for 40 gold.
```

```
== Farm City :: My Inventory ==
Welcome, TEST3!
Rank: Novice           Gold: 960

My Seeds:
1. 2 Bags of Papaya

[M]ain | [B]uy | [G]ift | Select choice >
```

3. A certain amount of gold is deducted according to the price of the seed.

PROJECT - Farm City

Alternate flow

- 2a. When the player's gold is not enough to pay for what he chooses, the System displays the information that the gold is not enough.

```
== Farm City :: My Inventory ==
Welcome, TEST3!
Rank: Novice                      Gold: 960

Seeds Available:
1. Papaya costs: 20 gold
   Harvests in: 30 mins
   XP Gained: 8
2. Pumpkin costs: 30 gold
   Harvests in: 60 mins
   XP Gained: 5
3. Sunflower costs: 40 gold
   Harvests in: 120 mins
   XP Gained: 20
4. Watermelon costs: 50 gold
   Harvests in: 240 mins
   XP Gained: 1

[M]ain | Select choice > 4
Enter quantity > 20
Your gold is not enough!

== Farm City :: My Inventory ==
Welcome, TEST3!
Rank: Novice                      Gold: 960

My Seeds:
1. 2 Bags of Papaya

[M]ain | [B]uy | [G]ift | select choice >
```

3.11Send Gift

Main flow

Prerequisite: The Farmer is already logged in

1. From "My Inventory" menu, the player enters G to go to "Send a Gift" menu.

```
Enter your choice > 3

== Farm City :: My Inventory ==
Welcome, Apple Ng!
Rank: Novice                      Gold: 530

My Seeds:
1. 5 Bags of Papaya
2. 1 Bags of Pumpkin
3. 1 Bags of Sunflower
4. 2 Bags of Watermelon

[M]ain | [B]uy | [G]ift | Select choice > G
```

PROJECT - Farm City

2. The System displays the seeds available for gifting.
3. The player chooses a seed to gift.
4. The player enter one or more recipient name's, defining "," between usernames for more than one recipient.
5. The system displays the message if gifting is successful.

```
[M]ain : [B]uy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice           Gold: 700

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > billy
Gift sent to billy
```

```
Send to > maria, kenneth
Gift sent to maria
Gift sent to kenneth
```

Alternate flow

- 4a. The player enter one or more recipient name's,

- i. The user has no seed in his inventory → The system informs the user, that user inventory is empty.

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

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > maria
The gift is not available in your inventory. Please try again
```

- ii. The user enters user's username, to send gift to own self → The system informs the user, that user cannot send gift to yourself.

PROJECT - Farm City

```
[M]ain : [Bluy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice                      Gold: 1000

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > apple
You cannot send gift to yourself.
```

- iii. The user enters a username not found in his friend list / user has no friends → The system informs the user, specifically which friend is not in his friend list.

```
[M]ain : [Bluy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice                      Gold: 1000

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > test
The gift is not available in your inventory. Please try again
```

- iv. The user tries to send a recipient twice in a day → The system informs the user, that user has already sent a gift to recipient for the day

```
[M]ain : [Bluy : [G]ift : Select choice > G
== Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice                      Gold: 800

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > billy, billy
Gift sent to billy
You have already sent a gift to billy today.
```

- v. The amount of recipient is more than the user choice of seeds → Following the user's order of recipients, the system only gift to the first few. The system informs the user, the success of gifting for each friend.

PROJECT - Farm City

```
-- Farm City :: My Inventory ==
Welcome, Apple Ng!
Rank: Novice                      Gold: 800

My Seeds:
i. 2 Bags of Papaya

[M]ain : [B]uy : [G]ift : Select choice > G
-- Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice                      Gold: 800

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > maria,billy,peter,jack
Gift sent to maria
Gift sent to billy
The gift is not available in your inventory. Please try again
The gift is not available in your inventory. Please try again
```

- vi. The user sends/has send more than 5 gifts in a day → The system sends gifts up to 5 gifts a day, once the user reaches maximum for the day, user cannot send gifts. The system informs the user, that the user has reached the maximum for the day.

```
[M]ain : [B]uy : [G]ift : Select choice > G
-- Farm City :: Send a Gift ==
Welcome, Apple Ng
Rank: Novice                      Gold: 800

Gifts Available:
1. 1 Bag of Papaya Seeds
2. 1 Bag of Pumpkin Seeds
3. 1 Bag of Sunflower Seeds
4. 1 Bag of Watermelon Seeds
[M]ain : Select choice > 1
Send to > jack,maria,billy,effendy,peter,kenneth
Gift sent to jack
Gift sent to maria
Gift sent to billy
Gift sent to effendy
Gift sent to peter
Cannot send to kenneth. Your gift sending amount have reached the maximum today.
```