



SMU

SINGAPORE MANAGEMENT
UNIVERSITY

Project DukeHunt

[IS 201] Object Oriented Application Development

Group Project Report – Final Submission

Submit Date:

8th April 2014

Instructor:

Prof David Lo

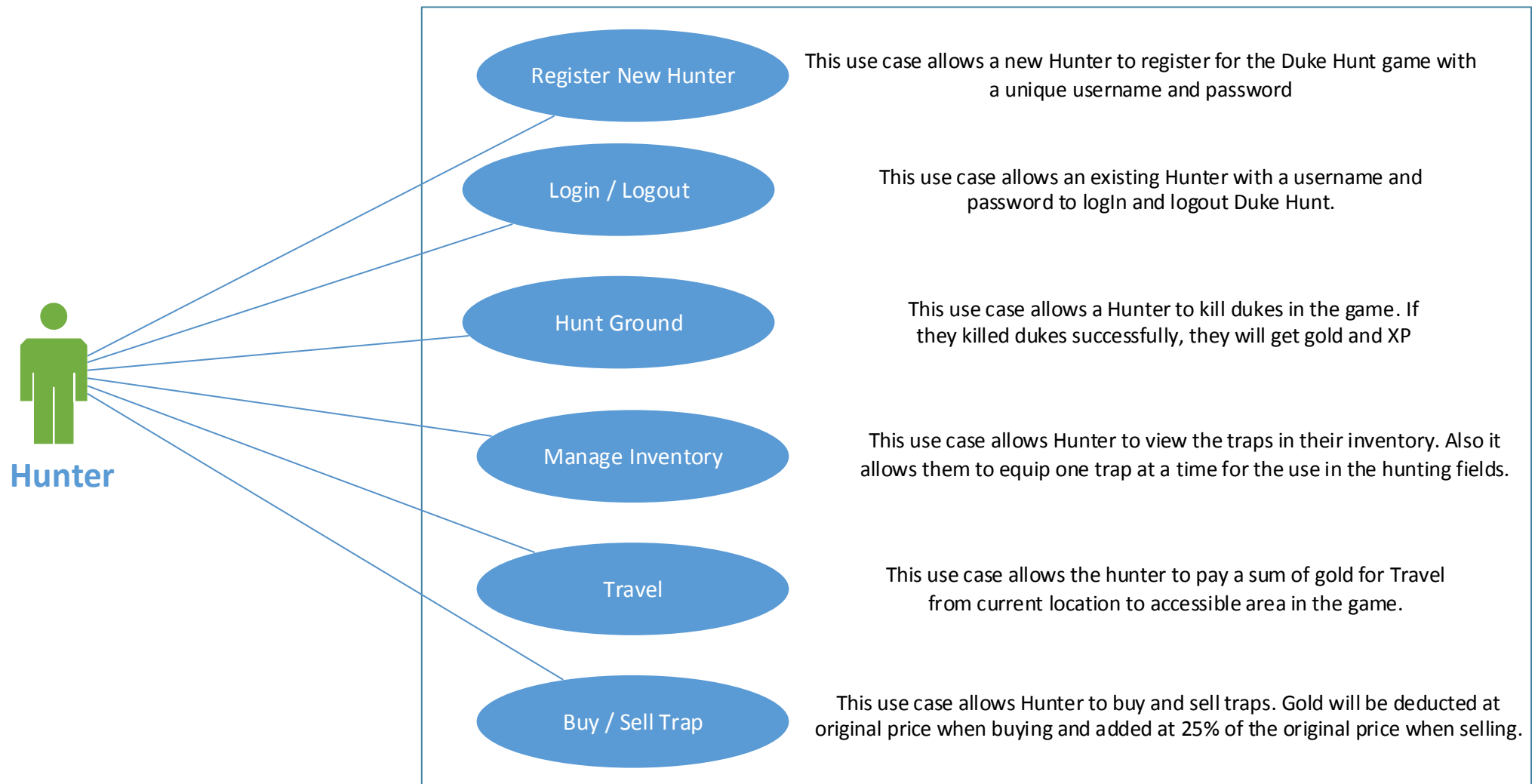
Produced by:

G4-T03 Tee An Ting | Yuan Yuxuan | Tanisha Sanwer

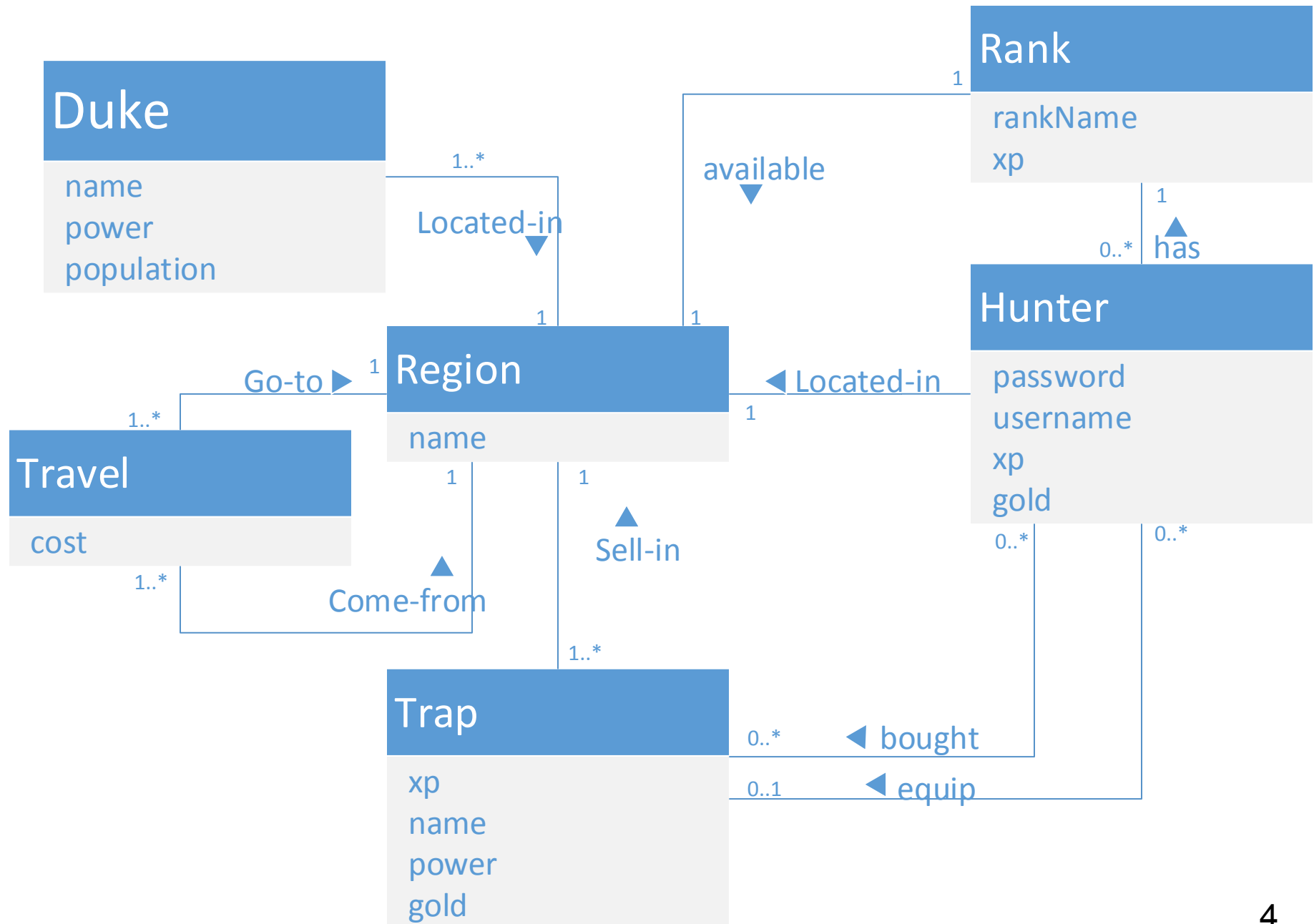
CONTENT PAGE

CHAPTER	PAGE
1. ANALYSIS AND DESIGN	
1.1 USE CASE DIAGRAM -----	3
1.2 DOMAIN DIAGRAM -----	4
1.3 CLASS DIAGRAM -----	5
1.4 USE CASE –REGISTER NEW HUNTER	
Use Case Scenarios -----	6
System Sequence Diagram -----	7
Sequence Diagram -----	8
1.5 USE CASE –LOGIN/LOGOUT	
Use Case Scenarios -----	9
System Sequence Diagram -----	10
Sequence Diagram -----	11
1.6 USE CASE –HUNTING GROUND	
Use Case Scenarios -----	12
System Sequence Diagram -----	13
Sequence Diagram -----	14
1.7 USE CASE –MANAGE INVENTORY	
Use Case Scenarios -----	15
System Sequence Diagram -----	16
Sequence Diagram -----	17
1.8 USE CASE –TRAVEL	
Use Case Scenarios -----	18
System Sequence Diagram -----	19
Sequence Diagram -----	20
1.9 USE CASE –BUY/SELL TRAP	
Use Case Scenarios -----	21
System Sequence Diagram -----	22
Sequence Diagram -----	23
1.10 OBJECT-ORIENTED DESIGN CONSIDERATIONS	24/25
2. MISCELLANEOUS	
2.1 CHALLENGES & TAKEAWAYS/LESSONS LEARNT -----	26
3. SCREENSHOTS	
Register New Hunter -----	27
Login/Logout -----	28
Hunting Ground(1) -----	29
Hunting Ground(2) -----	30
Manage Inventory -----	31
Buy/Sell Trap -----	32
Travel -----	33

1.1 Use case diagram and description



1.2 Domain Diagram



5



1.4 Use Case Scenario: Register New Hunter

Actor

- Hunter

Precondition

- New member without existing account

Main flow of events

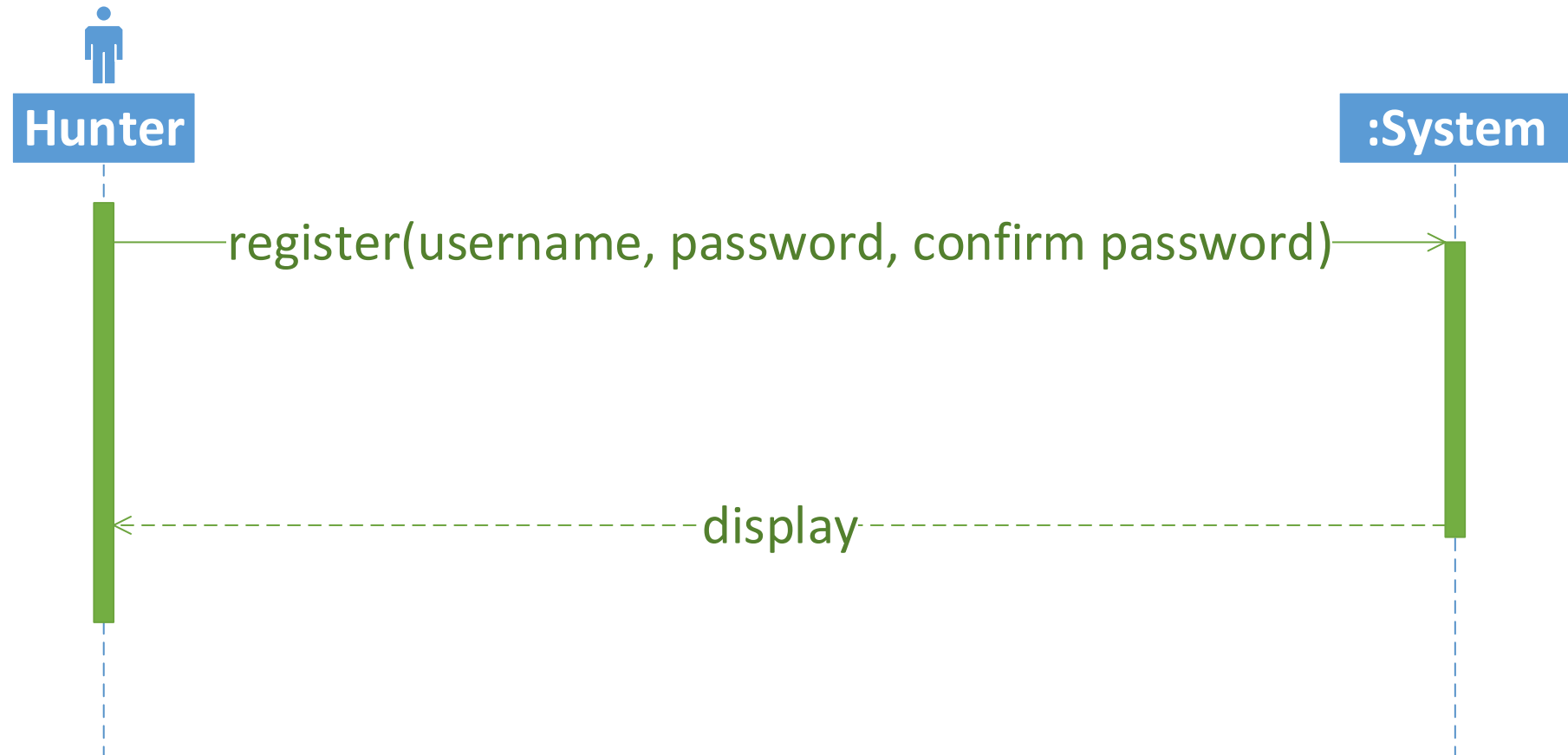
1. This use case begins when the hunter start the startup menu and choose “Register”.
2. The system shows the Registration Menu.
3. The hunter enters a “username”.
4. The system checks that the username is available. If it is, the system require the hunter to key in password.
5. The hunter keys in “password”.
6. The system requires the hunter to “confirm password”.
7. The hunter keys in “confirm password”.
8. The system check that two password entered are the same. If it is, the system creates an account for hunter and shows the success page, and end the registration.

Alternate flow of events

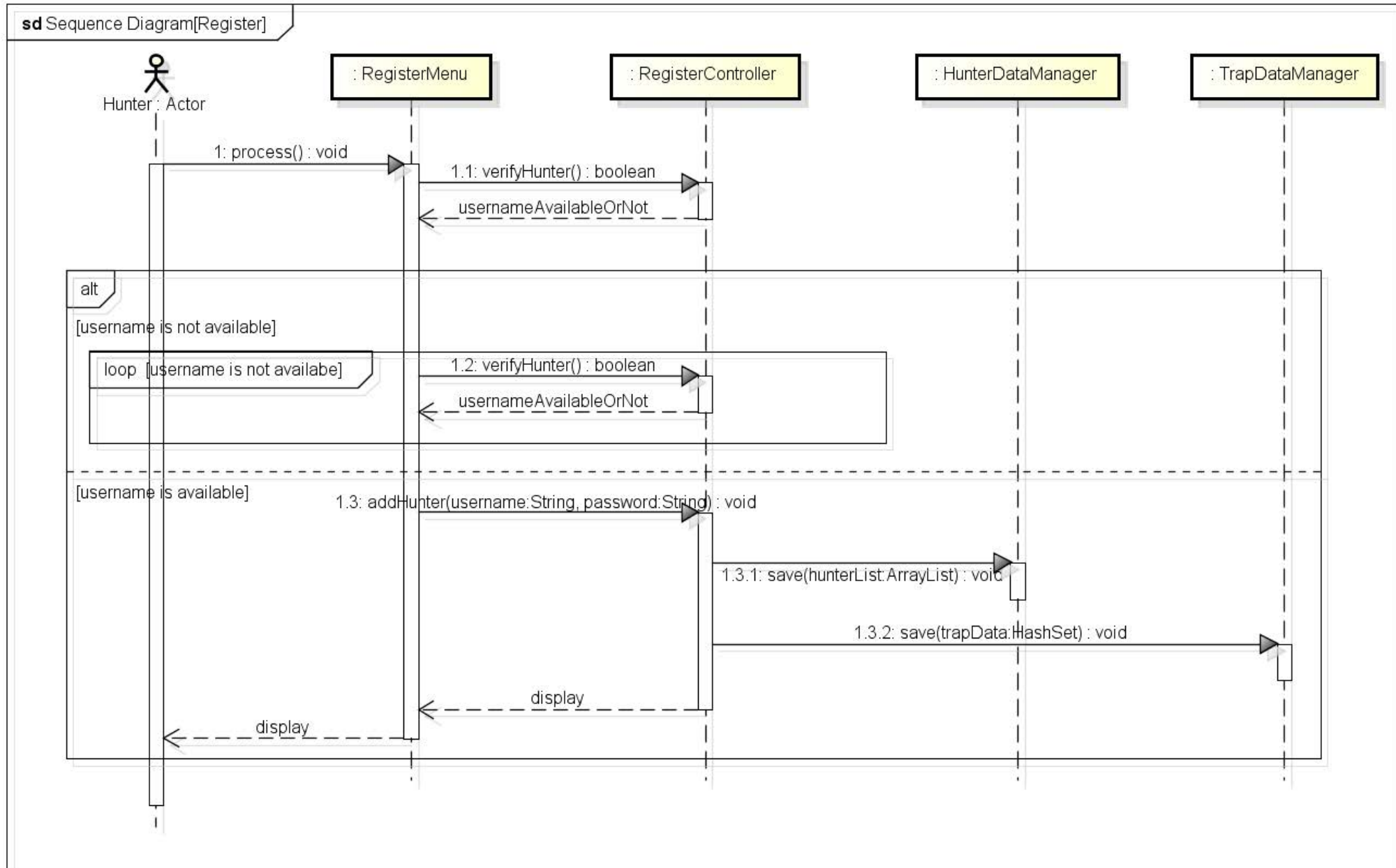
4a. Username has been in used

- i) The system shows the invalid information and requires the hunter to enter another “username”.
- ii) The hunter keys in another “username”, until the system accepts it.

1.4 SSD:Register New Hunter



1.4 SS:Register New Hunter



1.5 Use Case Scenario: Login/Logout

Actor

- Hunter

Precondition

- The hunter must have an existing account

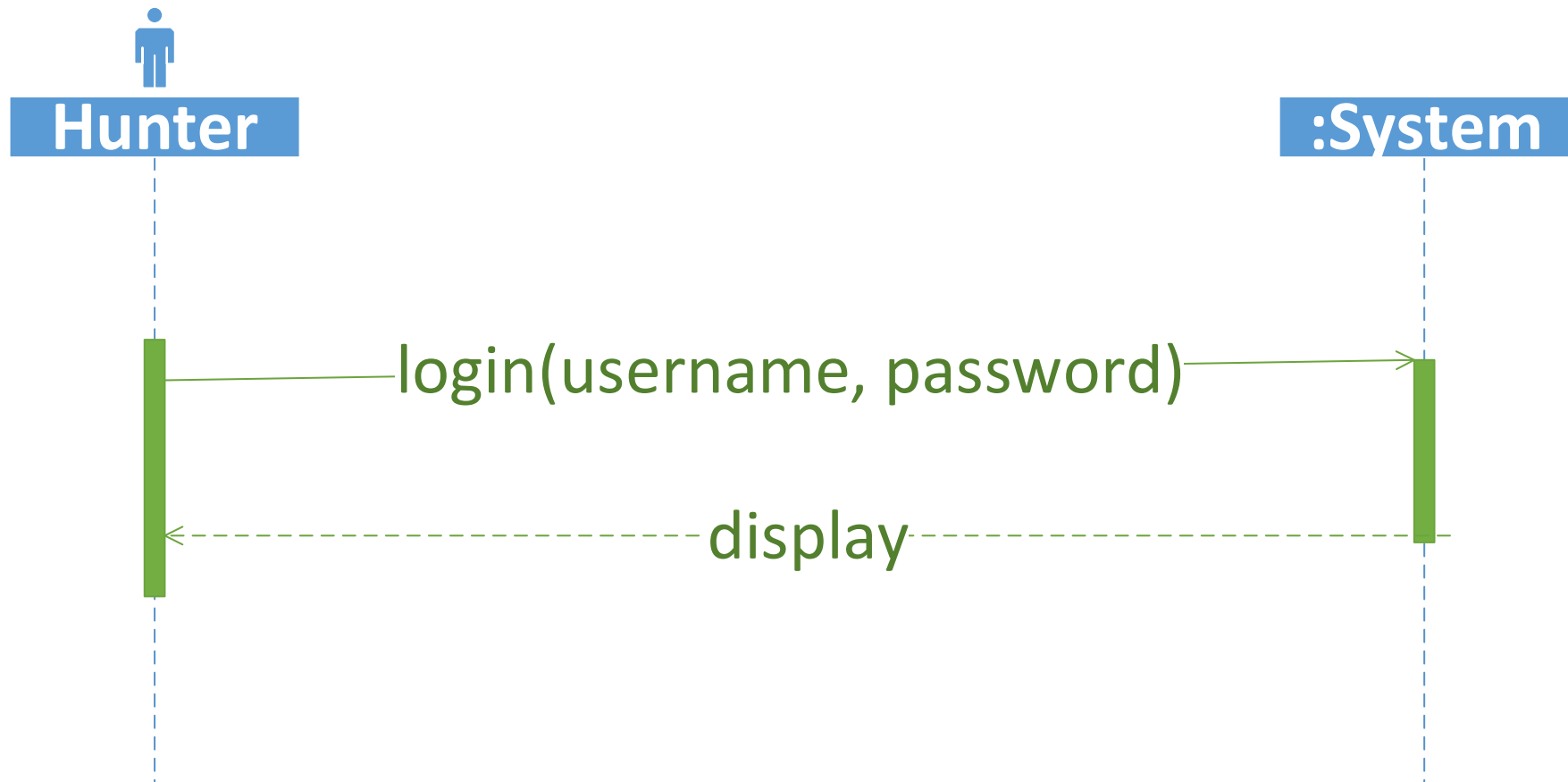
Main flow of events

1. This use case begins when the hunter start the start-up menu and choose “Login”.
2. The system shows the Login Menu.
3. The hunter enters the “username” and “password”.
4. The system checks that the username and password match with the database. If it is, the system will show the successful login page, and display the main menu.

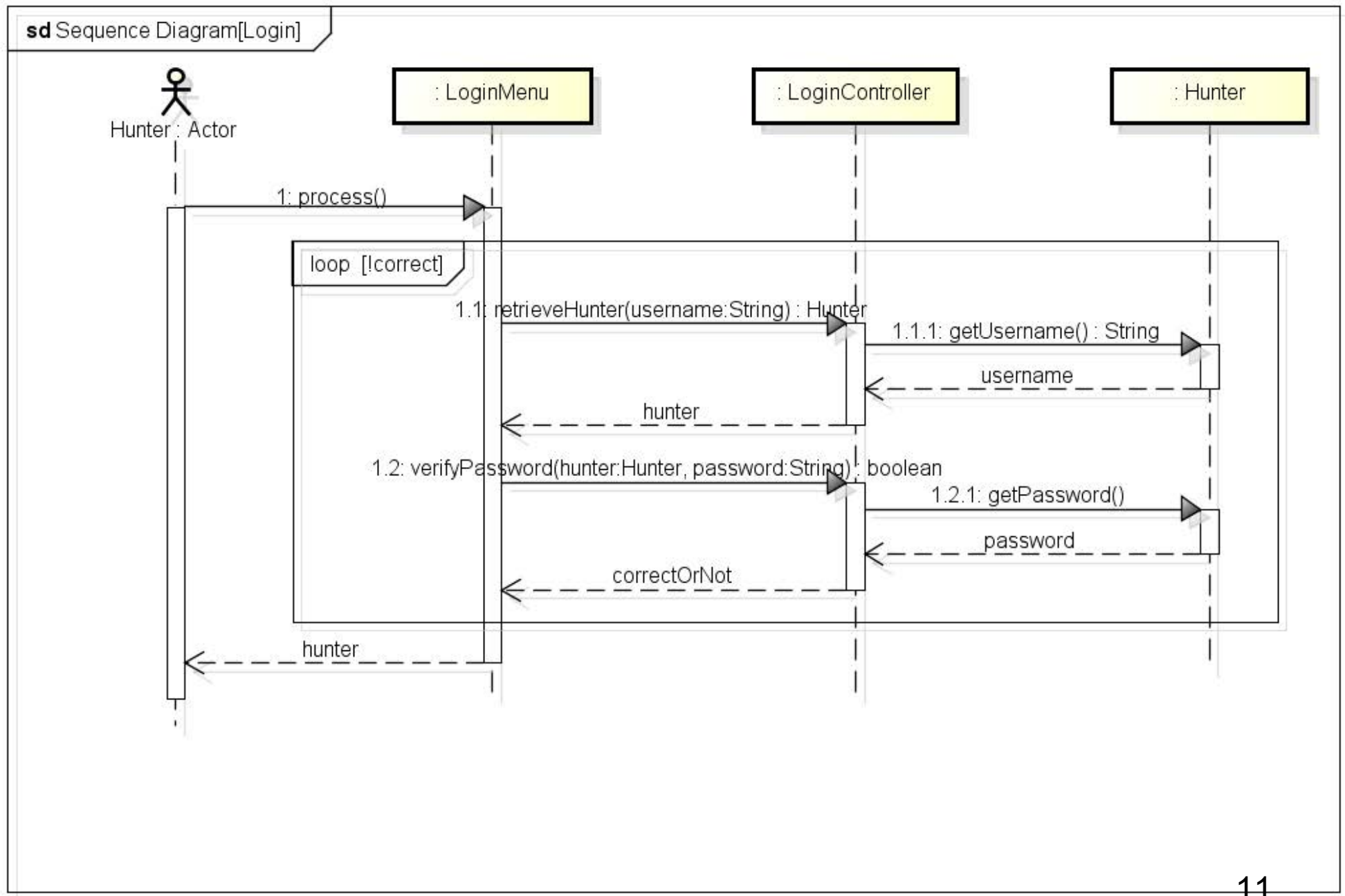
Alternate flow of events

- 4a. Username and password does not match with the system.
 - i) The system shows the invalid information and requires the hunter to key in the “username” and “password”.

1.5 SSD: Login/Logout



1.5 SS: Login/Logout



1.6 Use Case Scenario: Hunt Ground

Actor

- Hunter

Precondition

- The hunter is logged in

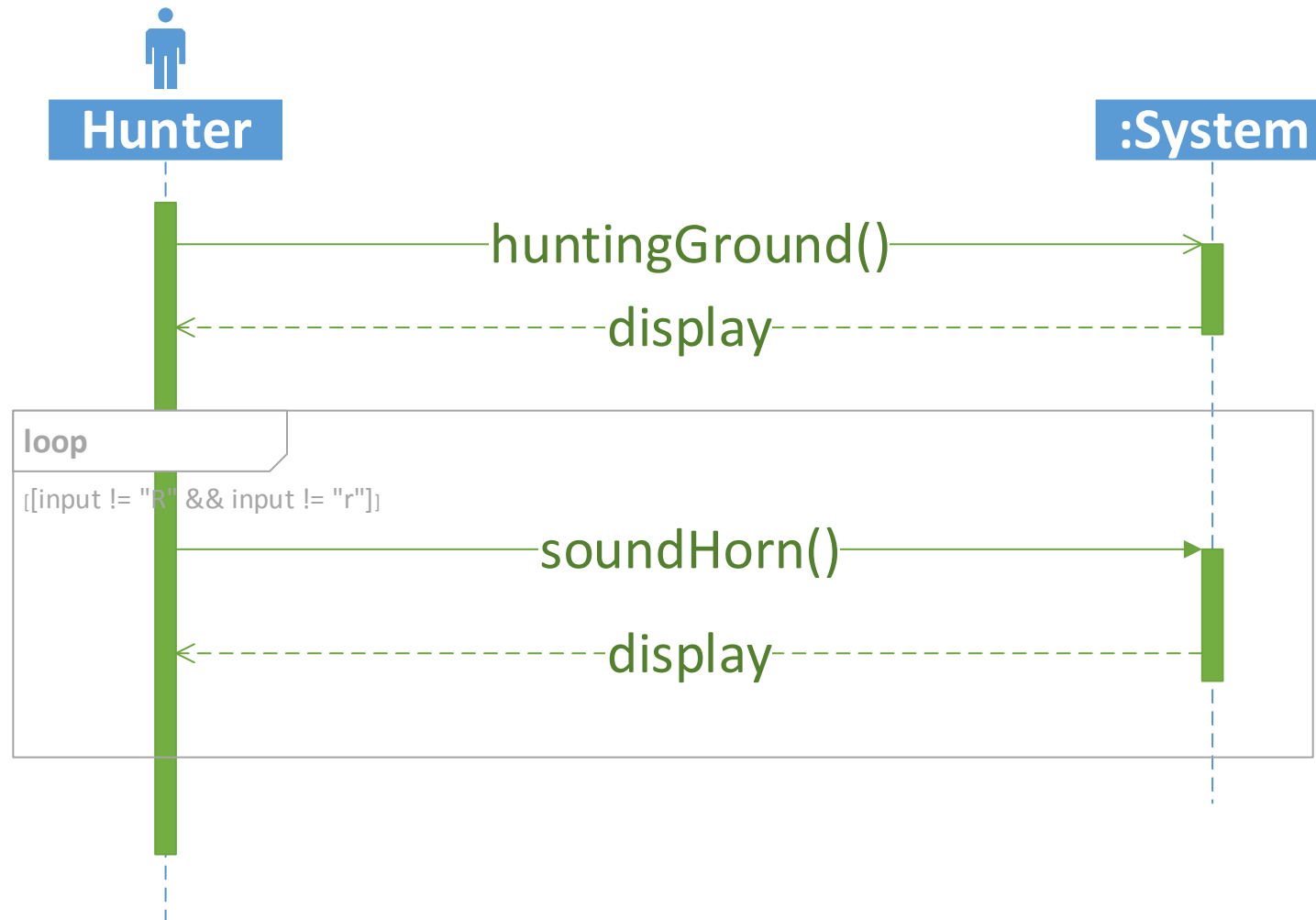
Main flow of events

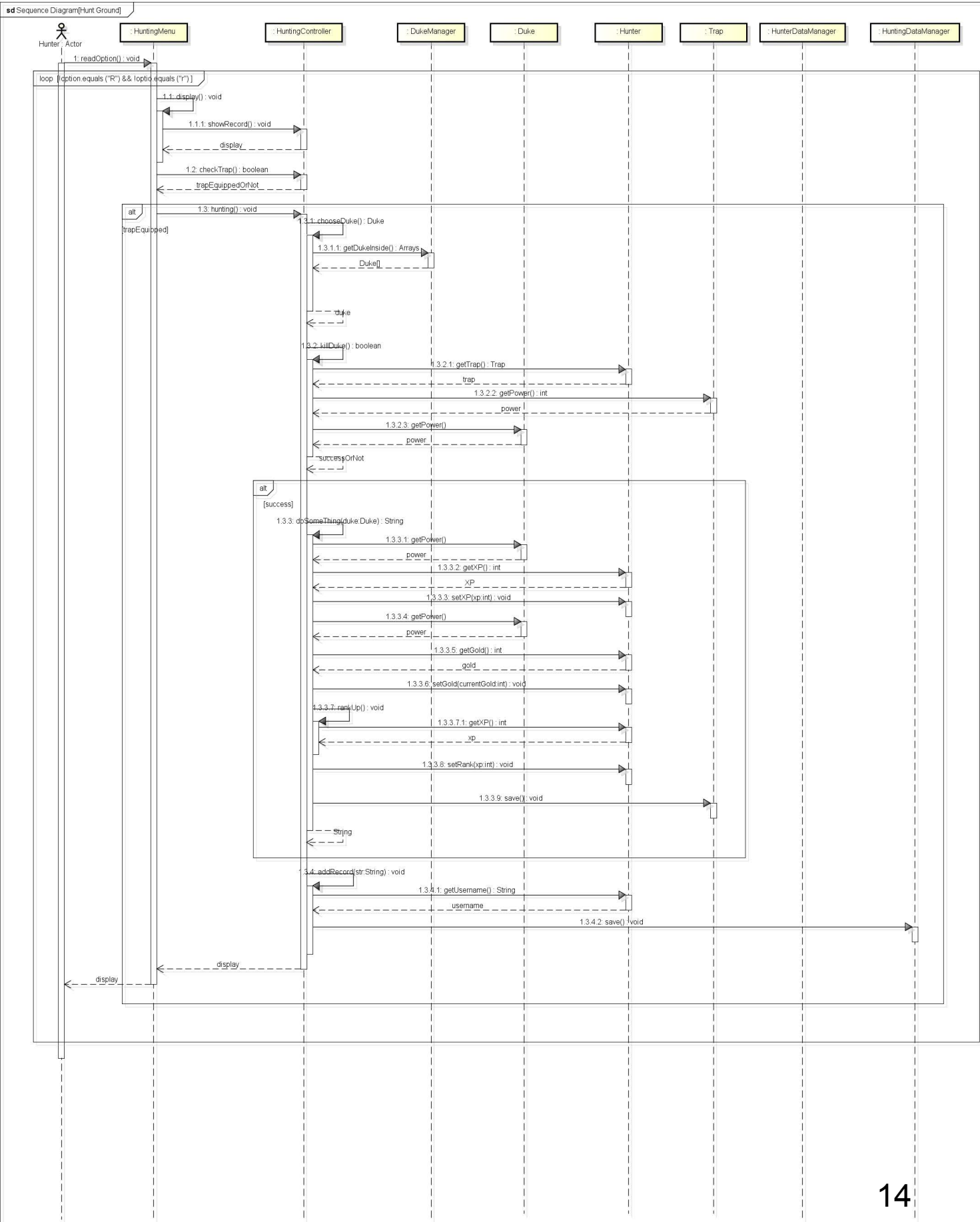
1. This use case begins when the hunter is on the main menu display screen.
2. The hunter enters “Hunting Ground”.
3. The system displays a log of the last 10 hunts of the Hunter.
4. The hunter can sound the horn to hunt.
5. The system will display the hunt details.
6. The hunter can repeat step 4 – 6.
7. The hunter ends the Hunting Ground.

Alternate flow of events

NA

1.6 SSD: Hunting Ground





1.7 Use Case Scenario: Manage Inventory

Actor

- Hunter

Precondition

- The hunter is logged in

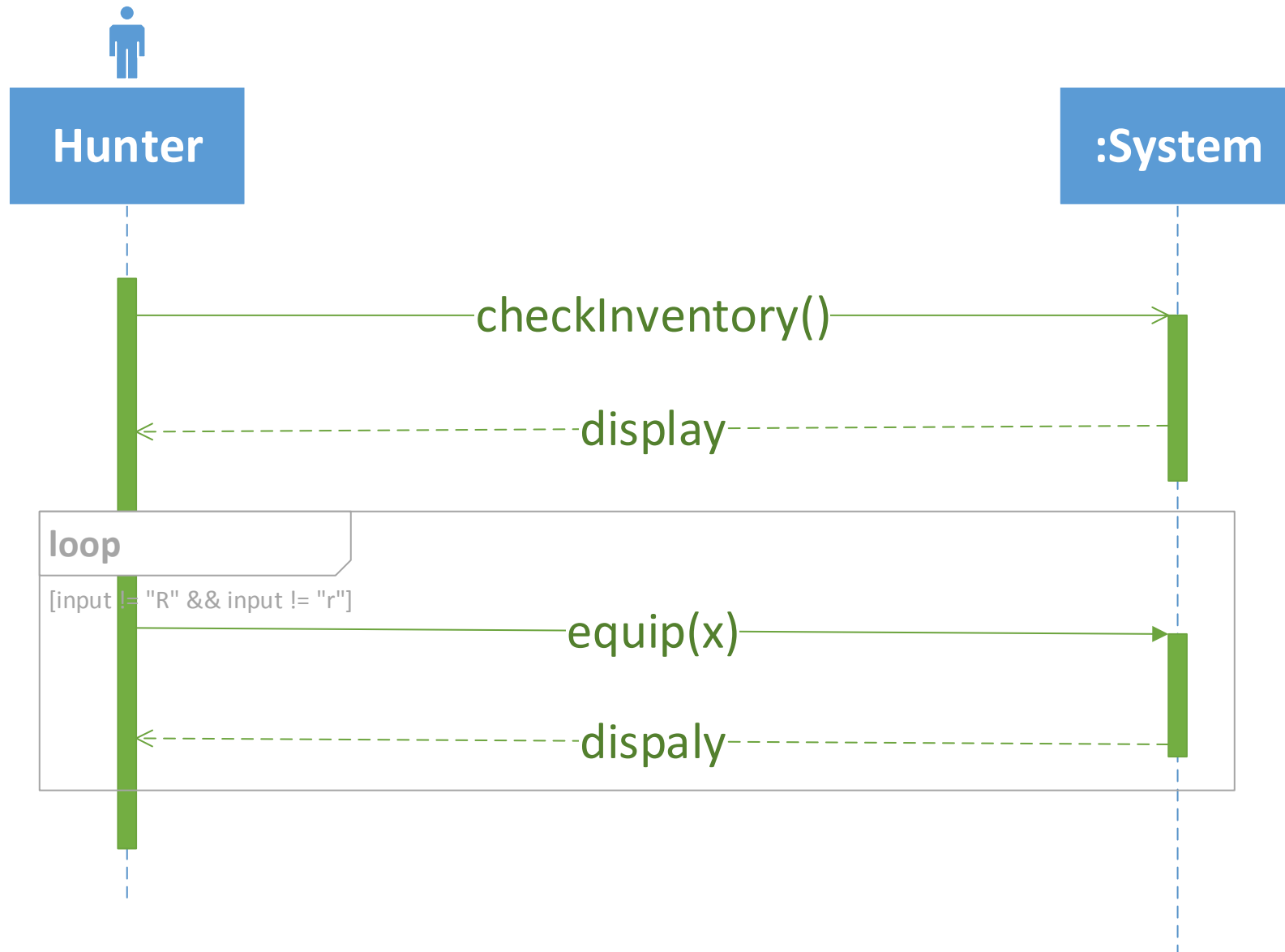
Main flow of events

1. This use case begins when the hunter is on the main menu display screen and chooses “My Inventory”.
3. The system displays the equipped trap and traps in inventory.
4. The hunter can select a trap in the inventory to equip.
5. The system changes the equipped trap for hunter, puts it into the inventory and displays the updated traps information
6. The hunter can repeat step 4 – 6.
7. The hunter ends the Manage Inventory use case.

Alternate flow of events

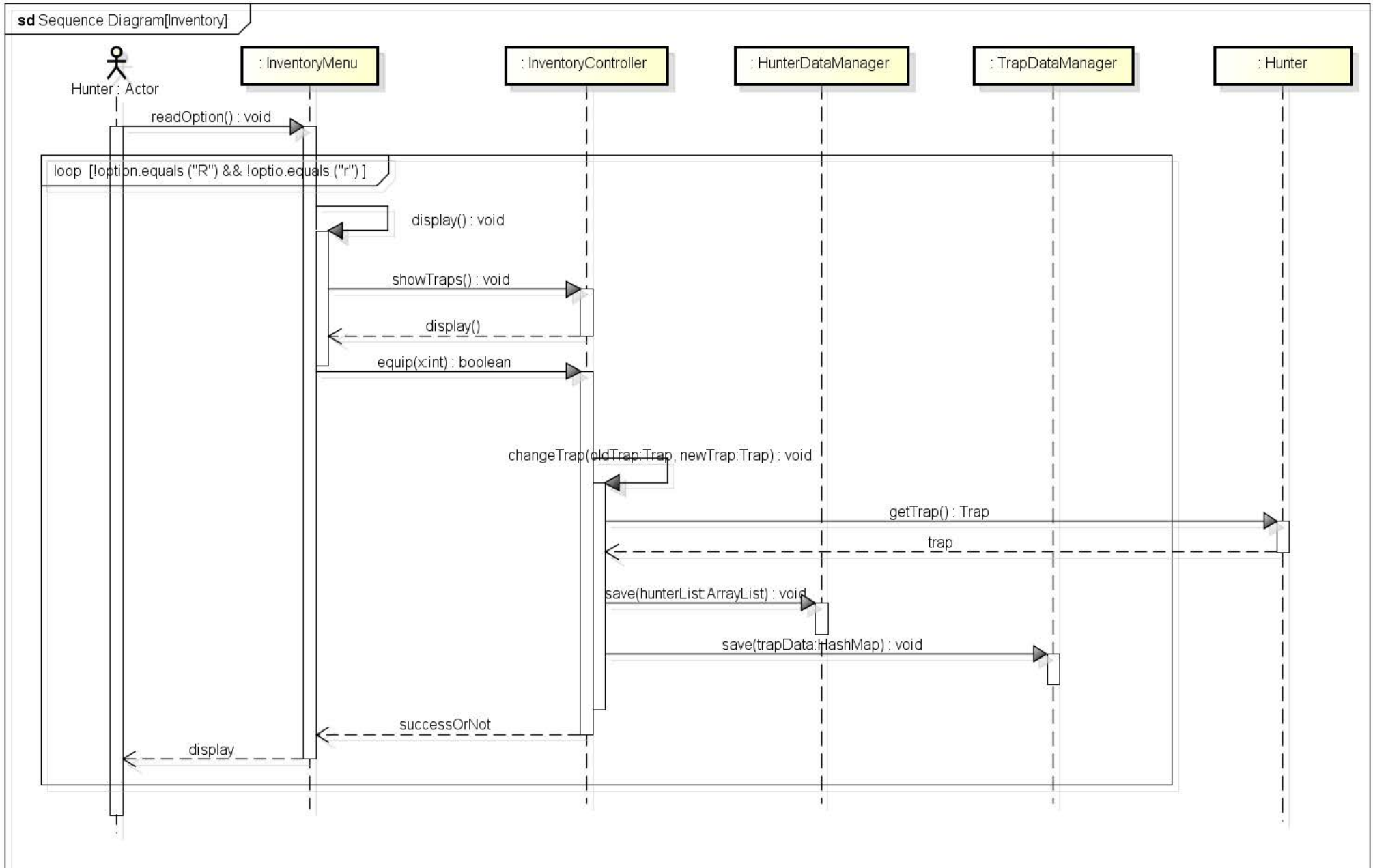
NA

1.7 SSD: Manage Inventory



* **X**: The integer that indicates the option of the hunter

1.7 SS: Manage Inventory



1.8 Use Case Scenario: Buy/Sell Trap

Actor

- Hunter

Precondition

- The hunter is logged in

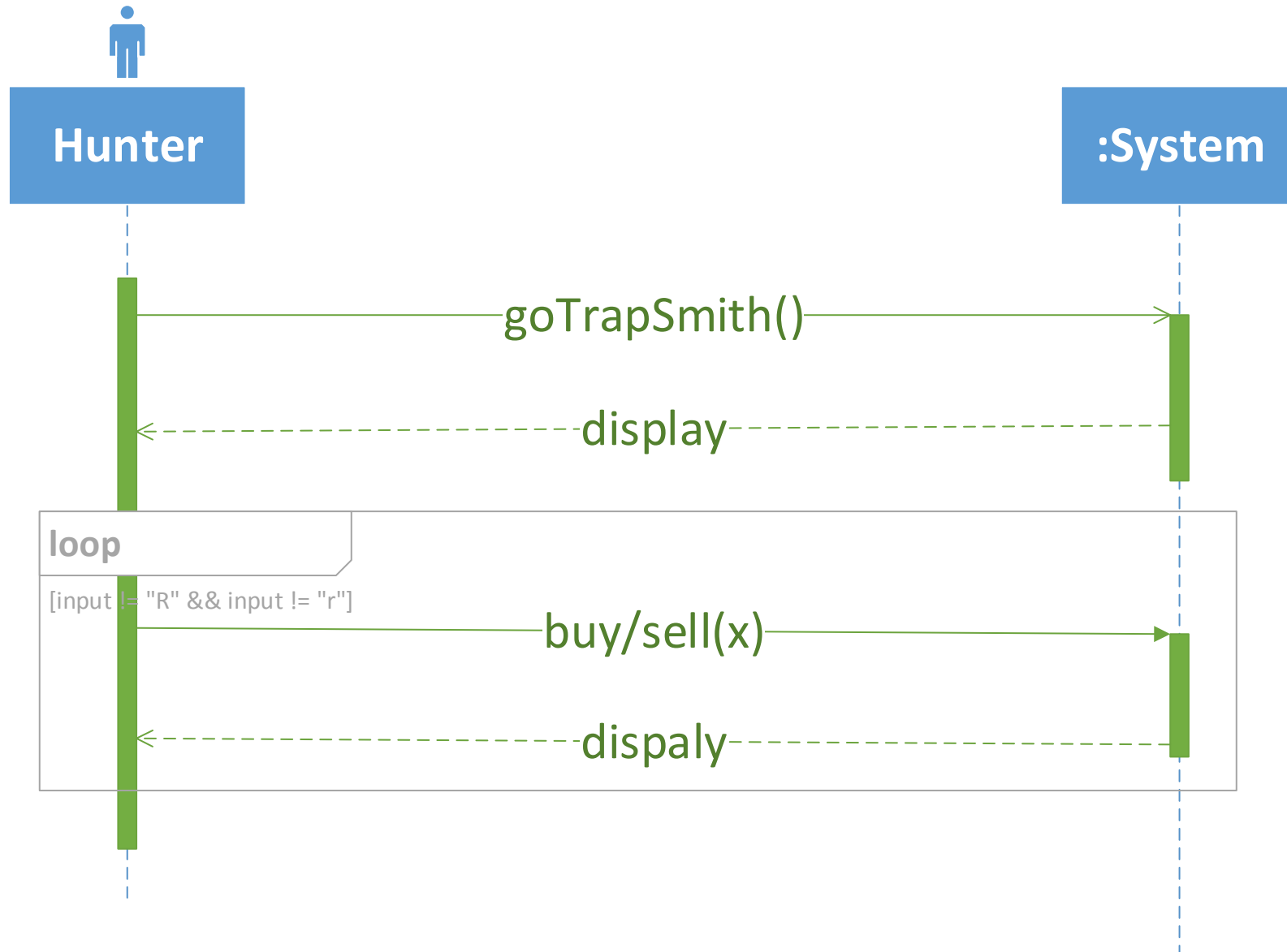
Main flow of events

1. This use case begins when the hunter is on the main menu display screen and chooses “The Trap Smith”.
2. The system displays the available traps for the region.
3. The hunter makes option to sell or buy a certain trap.
4. The system checks if the option is accessible. If it is, the system changes the trap status and deducts or adds corresponding gold for the hunter.
5. The hunter can repeat step 3 – 5.
6. The hunter ends the Buy / Sell Trap use case.

Alternate flow of events

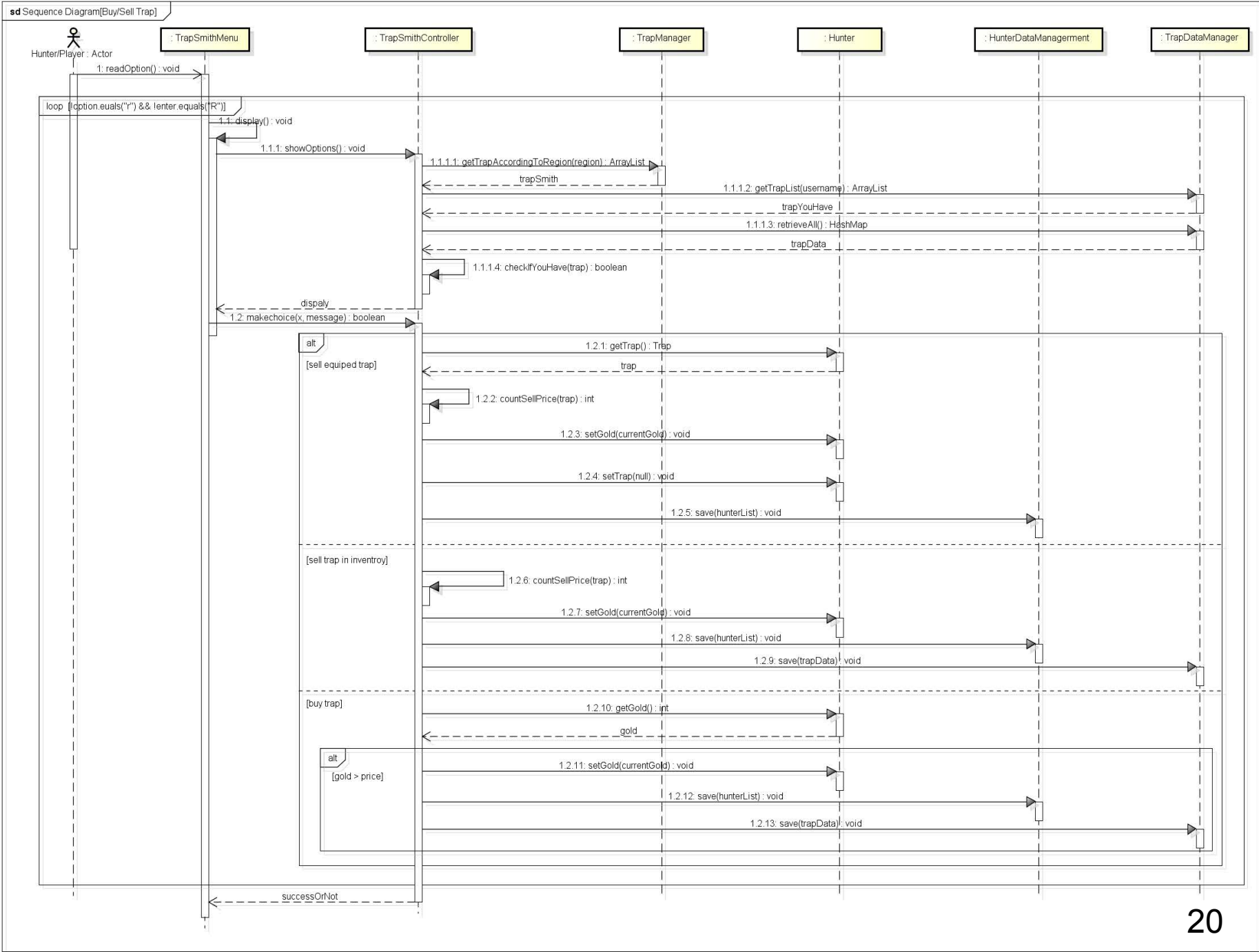
- 3a. The hunter chooses to sell his or her last trap
 - i) The system shows the invalid information and display the TrapSmith menu again.
- 3b. The hunter does not have enough money/xp to buy the trap.
 - ii) The system shows the invalid information and display the TrapSmith menu again.

1.8 SSD: Buy/Sell Trap



* **X**: The integer that indicates the option of the hunter

1.8 SS: Buy/Sell Trap



1.9 Use Case Scenario: Travel

Actor

- Hunter

Precondition

- The hunter is logged in

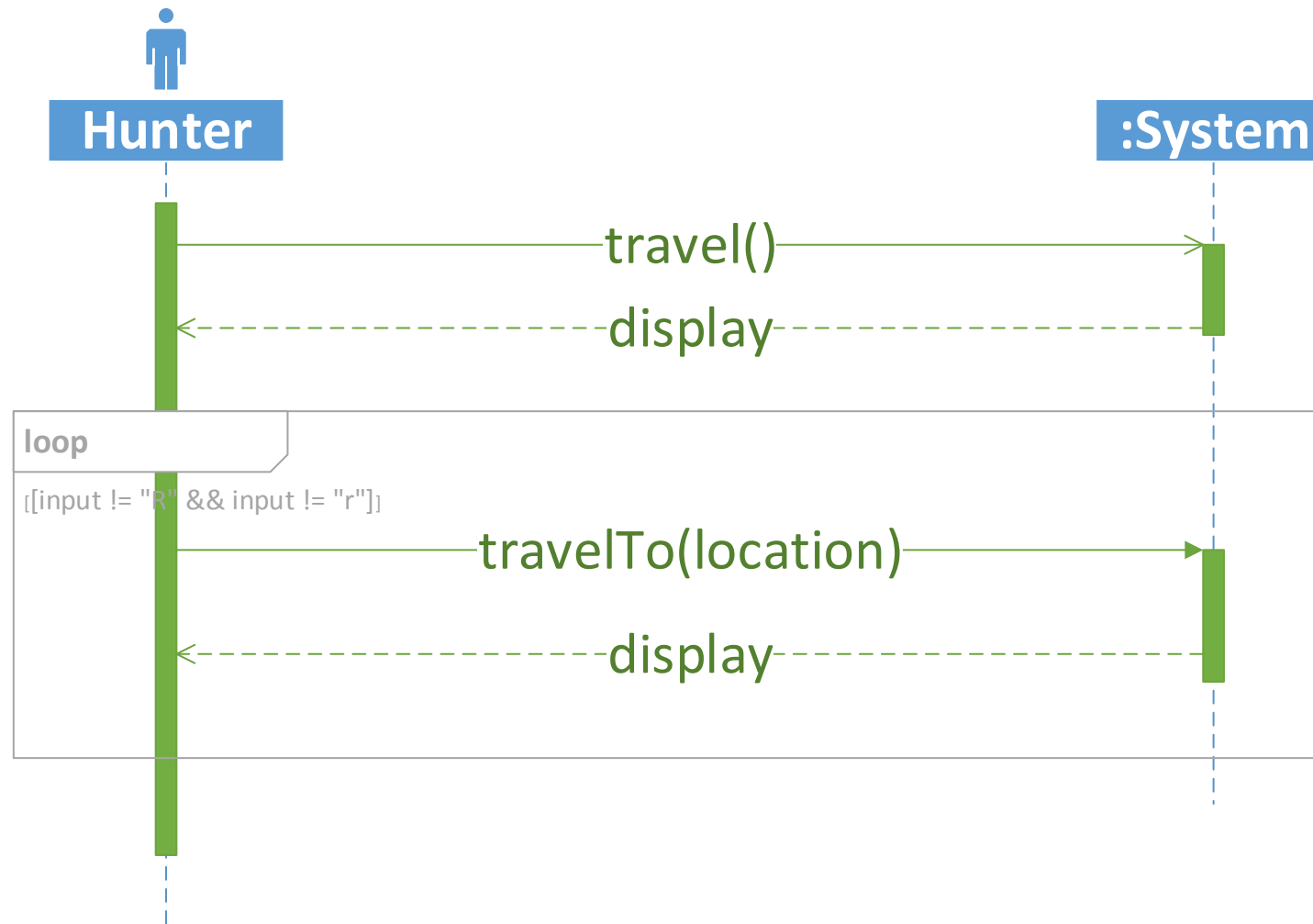
Main flow of events

1. This use case begins when the hunter is on the main menu display screen.
2. The hunter enters “Travel”.
3. The system displays the available areas for the travel.
4. The hunter makes option to travel to another region.
5. The system checks if the option is accessible according to hunter’s gold amount. If it is, the system changes the region, deducts corresponding gold for the hunter and displays the updated travel information.
6. The hunter can repeat step 4 – 6.
7. The hunter ends the Travel.

Alternate flow of events

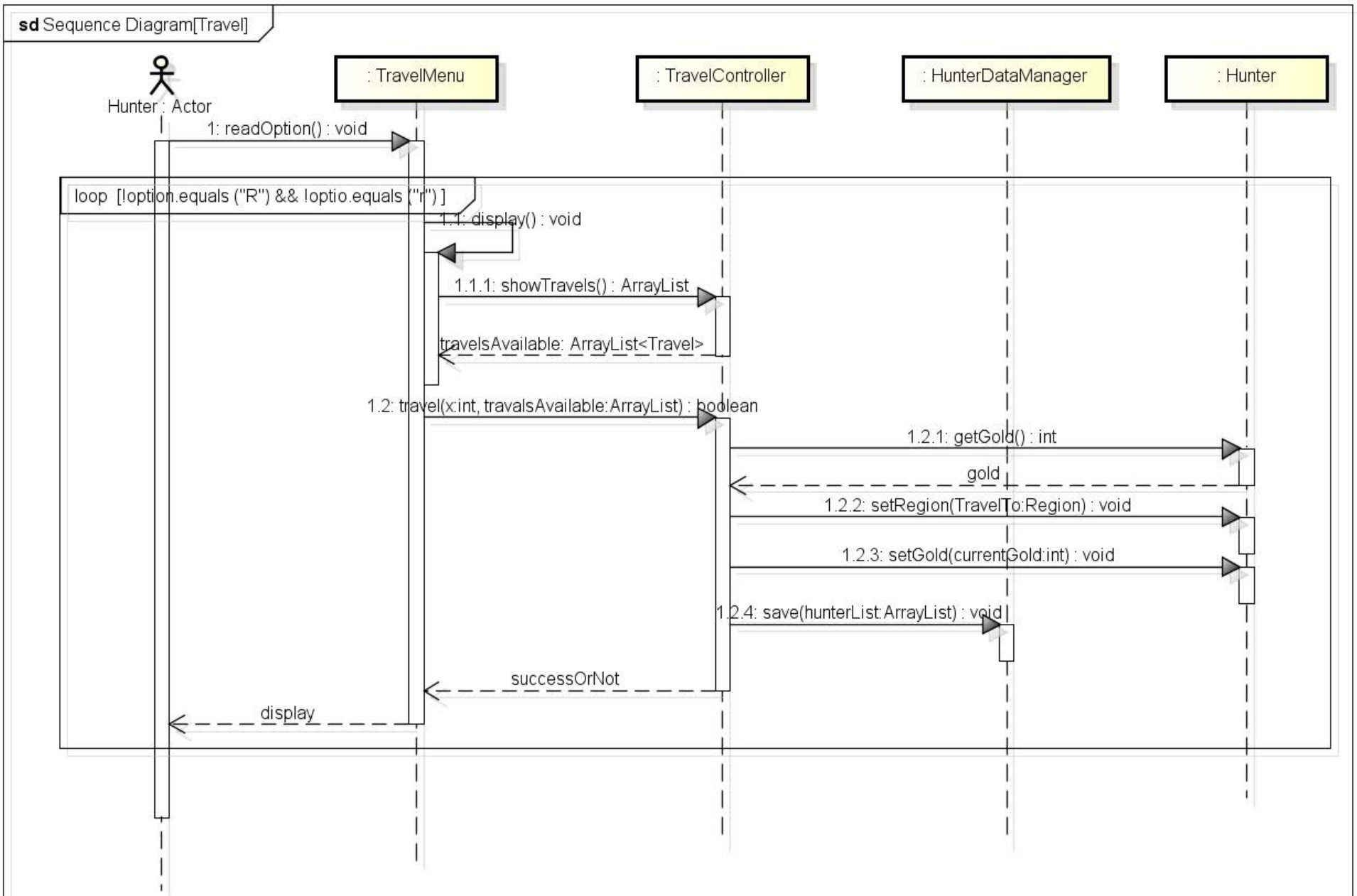
NA

1.9 SSD: Travel



* **X**: The integer that indicates the option of the hunter

1.9 SS: Travel



1.10 Object-oriented Design Considerations

We took 2 main approaches when considering the design of our program. One of which is the Model-View-Controller software architecture method. This means that our program can be organized based on several functionalities. In our packaging, we have introduced 6 individual packages (the last, application, runs the program), the Entities, Exceptions and Data-Manager classes are predominantly of the Entity stereotype. The controllers are of the Controller stereotype and interact with the Data-Manager classes to update the databases. Finally the menu package contains menu classes that feature the user interface (UI) and the Menu-Controllers, which direct users to other menus, and interact with the Controller classes. This means that there are classes of the controller and boundary stereotype in this last package. We used this approach because it offers a lot of flexibility and allows us to scale the program with little problems. For example, when we wanted to do a Travel, we can interact with the Travel Controller, calling areas from the database the Travel Data Manager, and at the same time simply calling the Hunter Controller to update the Hunter's attributes (XP and gold) in the Hunter Database. This shows easy scalability of the program. Every function interacts with one central system.

Another approach that we took during development, which is common when using an Object-Oriented Programming (OOP) language, is Responsibility-Driven Design (RDD). This is the most preferable to plan the coding procedure because it helps to identify and allocate separate responsibilities to discrete objects. To know the responsibilities of an object will include having the knowledge about its private encapsulated data, meaning the attributes and what are its related objects and how do they interact with each other, as determined by the domain diagram. We assign responsibilities to the object based on its responsibility within the scope of the program. We also are able to determine when to create, modify and allow objects to navigate throughout the system.

During the project development, we ensure that responsibilities are separate when we code. This means that we enforce a loosely coupled development timeline. During integration, when one functionality contains bugs or does not work, it does not affect the whole program: only that functionality has to be explored to debug. During the development, when we implement

changes, the problems introduced are also only localized: meaning changes to not ripple and affect other separate parts of the program. It also incorporated with the Single Responsibility Principle where each class having its own responsibility. The advantage in this case will be to make the code easy to main with small, highly modularized code that is easy to understand and follow. Lastly, it has fine granularity of highly related functionality supports increased reuse potential of codes. The disadvantage of multiple responsibilities is that it will be hard to comprehend, reuse, maintain and affected by constant changes. This means that maintaining the code in the future will not be easy. Also, code is not easily being able to be reused in functionalities with similar algorithms/patterns. For example, similar code can be used to find buy/sell traps and also Travel. With these designs, it could overcome complex system that requires changes, reduces the system to break easily and require more changes, increase reusability of codes, working the right ways, prevent the system from being complicated and needless repetition of structures.

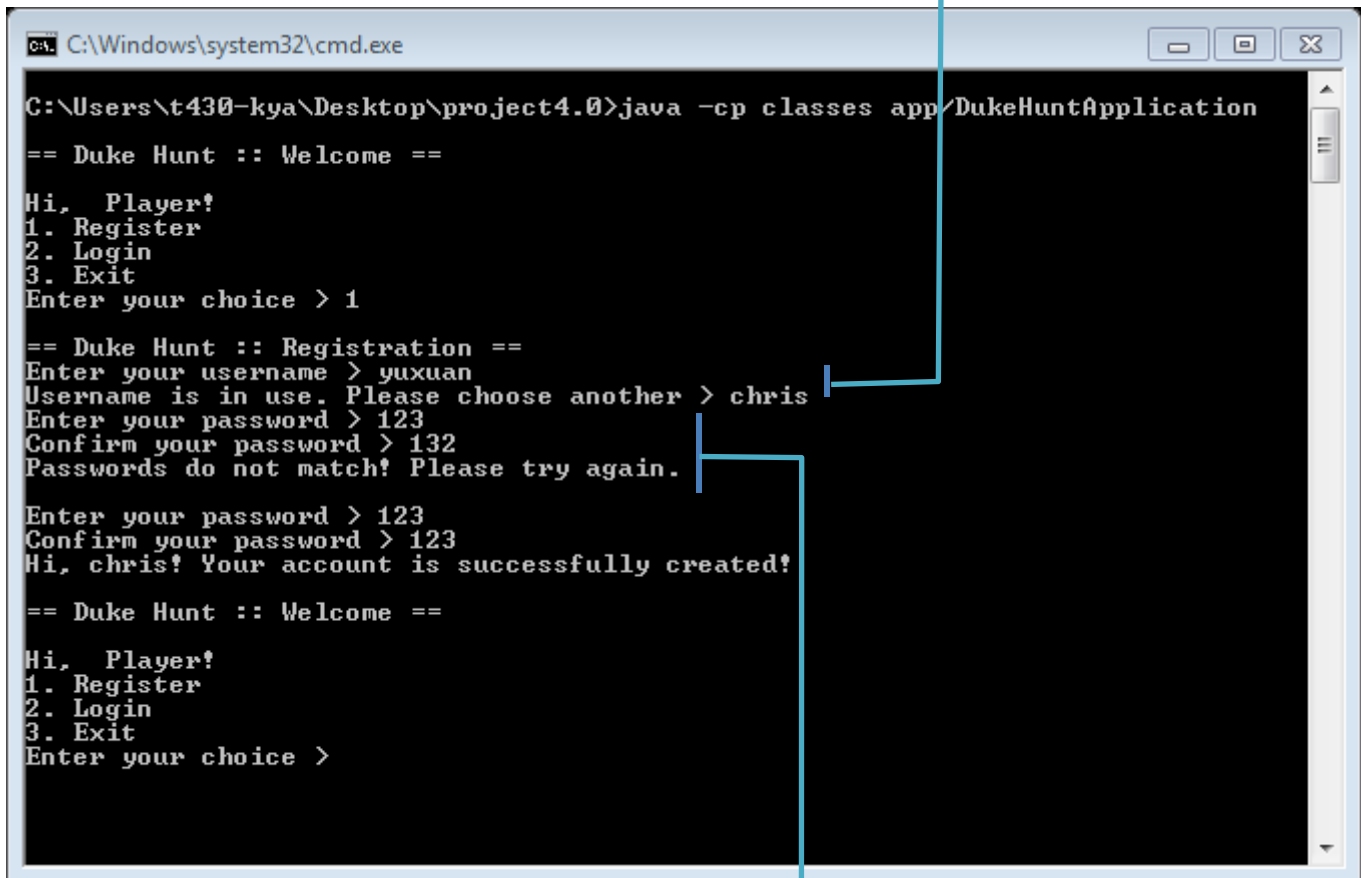
1.10 Challenges & Takeaways Learnt

Domains	Learning Outcomes	Challenges
Knowledge Application	The group project enhanced our knowledge learned in class and also improves our abilities to use Java language and understanding of Object-oriented design.	It is easy to understand the concepts but difficult to apply. We took some roundabout courses at first because of the misunderstanding about Object-oriented design principle so we had to change a lot of code after the interim consultation.
Collaboration	We helped and learned a lot from each other. Everybody contributed in different ways for the same goal. Through this group project, we acquired better collaborative skills and also made good friends.	Because of the heavy work-load in SMU, sometimes some members found difficulty to get time to do the project according to the plan. Even though we split the job to individual, people with more free time assisted “busy people” to do part of their work voluntarily.
Independent Learning	To make our project better, we tried some classes, methods beyond what we learnt in class. We searched a lot from API and online forum, discussed the project with other groups and also consulted from seniors and professors. We improved our ability to learn ourselves and search for external information efficiently and effectively.	Sometimes we cannot find the function we need at the first time. We tried to find alternate ways to tackle with the same problems. Because we were not skilled enough in applying the new functions we learnt ourselves, we need to be more patient and do more research when debugging.
Integration of requirements (business criteria) and technology	We learnt how to apply our IT knowledge and object-oriented development models to real-life functions. Even though what we are doing now is still very simple, with the accumulation of more IT knowledge and skill, I believe we can do more advanced, convent and good-look applications in the future. I am looking forward to SE, FYP and other more projects to improve our IT skill.	

3. Screenshots

Register New Hunter use case

If the username has been registered in the database, the system will request the hunter to choose another username.



```
C:\Windows\system32\cmd.exe

C:\Users\t430-kya\Desktop\project4.0>java -cp classes app/DukeHuntApplication

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice > 1

== Duke Hunt :: Registration ==
Enter your username > yuxuan
Username is in use. Please choose another > chris
Enter your password > 123
Confirm your password > 132
Passwords do not match! Please try again.

Enter your password > 123
Confirm your password > 123
Hi, chris! Your account is successfully created!

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice >
```

If the password and confirmed password do not match, the system will request the user to re-enter password and confirm password

Login/out use case

```
C:\Windows\system32\cmd.exe

C:\Users\t430-kya\Desktop\project4.0>java -cp classes app/DukeHuntApplication

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice > 2
Note: No hunters currently in the system.

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice >
```

If there is no hunter account in the database, the system will show the warning message and go back to the startup menu.

```
C:\Windows\system32\cmd.exe

C:\Users\t430-kya\Desktop\project4.0>java -cp classes app/DukeHuntApplication

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice > 2
Enter your username > tanisha
Enter your password > 123
You have entered an incorrect username/password.

Enter your username > yuxuan
Enter your password > 123

== Duke Hunt :: Main Menu ==
Welcome, Freshman yuxuan!
Location: SIS
XP: 0
Gold: 500

1. Hunting Ground
2. My Inventory
3. The Trap Smith
4. Travel
5. Logout
Enter your choice > 5
Bye bye!

== Duke Hunt :: Welcome ==

Hi, Player!
1. Register
2. Login
3. Exit
Enter your choice > 3
Thank you for playing Duke Hunt! Goodbye!
```

If the username does not exist in the database or the username and password do not match, the system will show the warning message and request the hunter to enter again.

Hunt Ground use case(1)

G4-T03

```

C:\Windows\system32\cmd.exe

== Duke Hunt :: Main Menu ==
Welcome, Junior yuxuan!
Location: SIS
XP: 7030
Gold: 5090

1. Hunting Ground
2. My Inventory
3. The Trap Smith
4. Travel
5. Logout
Enter your choice > 1

== Duke Hunt :: Hunting Ground ==
06-04-14 20:28:24 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained100 gold and81XP.

06-04-14 20:28:24 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.

06-04-14 20:28:23 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained32 gold and447XP.

06-04-14 20:28:23 @ SIS
A Greenhorn Duke escaped from my Trap of Least Resistance.

06-04-14 20:28:23 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.

06-04-14 20:28:23 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.

06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained61 gold and160XP.

06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained48 gold and73XP.

06-04-14 20:28:14 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained93 gold and459XP.

06-04-14 20:28:13 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.

[R]eturn to main : [S]ound horn >

```

If the hunter have 10 or more hunting records, the system will display the most recent 10 hunting records when the Hunter selects "Hunting Ground".

```

C:\Windows\system32\cmd.exe

== Duke Hunt :: Hunting Ground ==
[R]eturn to main : [S]ound horn > s

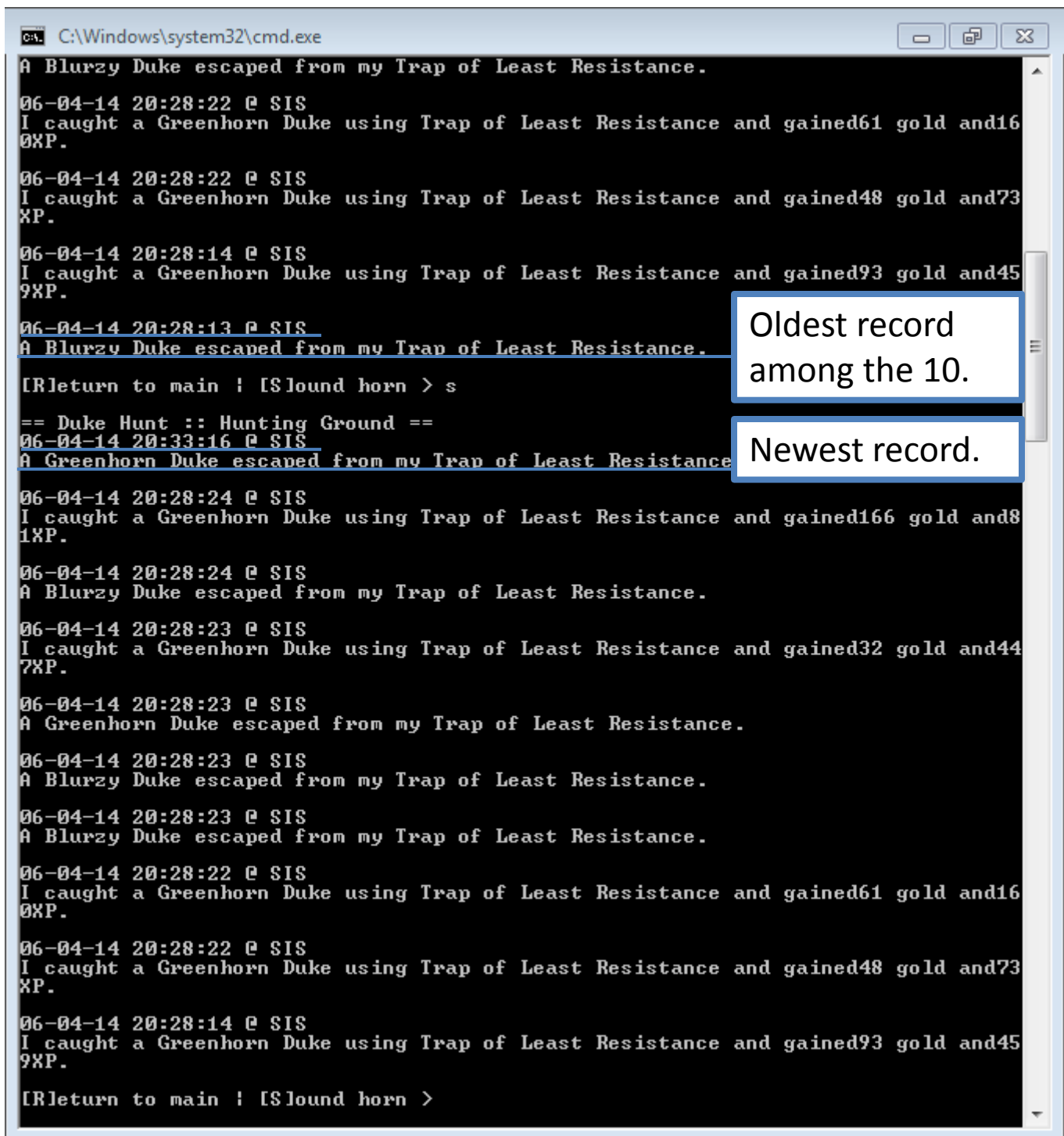
== Duke Hunt :: Hunting Ground ==
06-04-14 20:36:58 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained102 gold and363XP.

[R]eturn to main : [S]ound horn >

```

If the hunter does not have 10 or more hunting records, the system will display all the records by the hunter.

Hunt Ground use case(2)



```
C:\Windows\system32\cmd.exe

A Blurzy Duke escaped from my Trap of Least Resistance.
06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained61 gold and16
0XP.
06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained48 gold and73
XP.
06-04-14 20:28:14 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained93 gold and45
9XP.
06-04-14 20:28:13 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.

[Return to main : [Sound horn > s

== Duke Hunt :: Hunting Ground ==
06-04-14 20:33:16 @ SIS
A Greenhorn Duke escaped from my Trap of Least Resistance
06-04-14 20:28:24 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained166 gold and8
1XP.
06-04-14 20:28:24 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.
06-04-14 20:28:23 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained32 gold and44
7XP.
06-04-14 20:28:23 @ SIS
A Greenhorn Duke escaped from my Trap of Least Resistance.
06-04-14 20:28:23 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.
06-04-14 20:28:23 @ SIS
A Blurzy Duke escaped from my Trap of Least Resistance.
06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained61 gold and16
0XP.
06-04-14 20:28:22 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained48 gold and73
XP.
06-04-14 20:28:14 @ SIS
I caught a Greenhorn Duke using Trap of Least Resistance and gained93 gold and45
9XP.

[Return to main : [Sound horn >
```

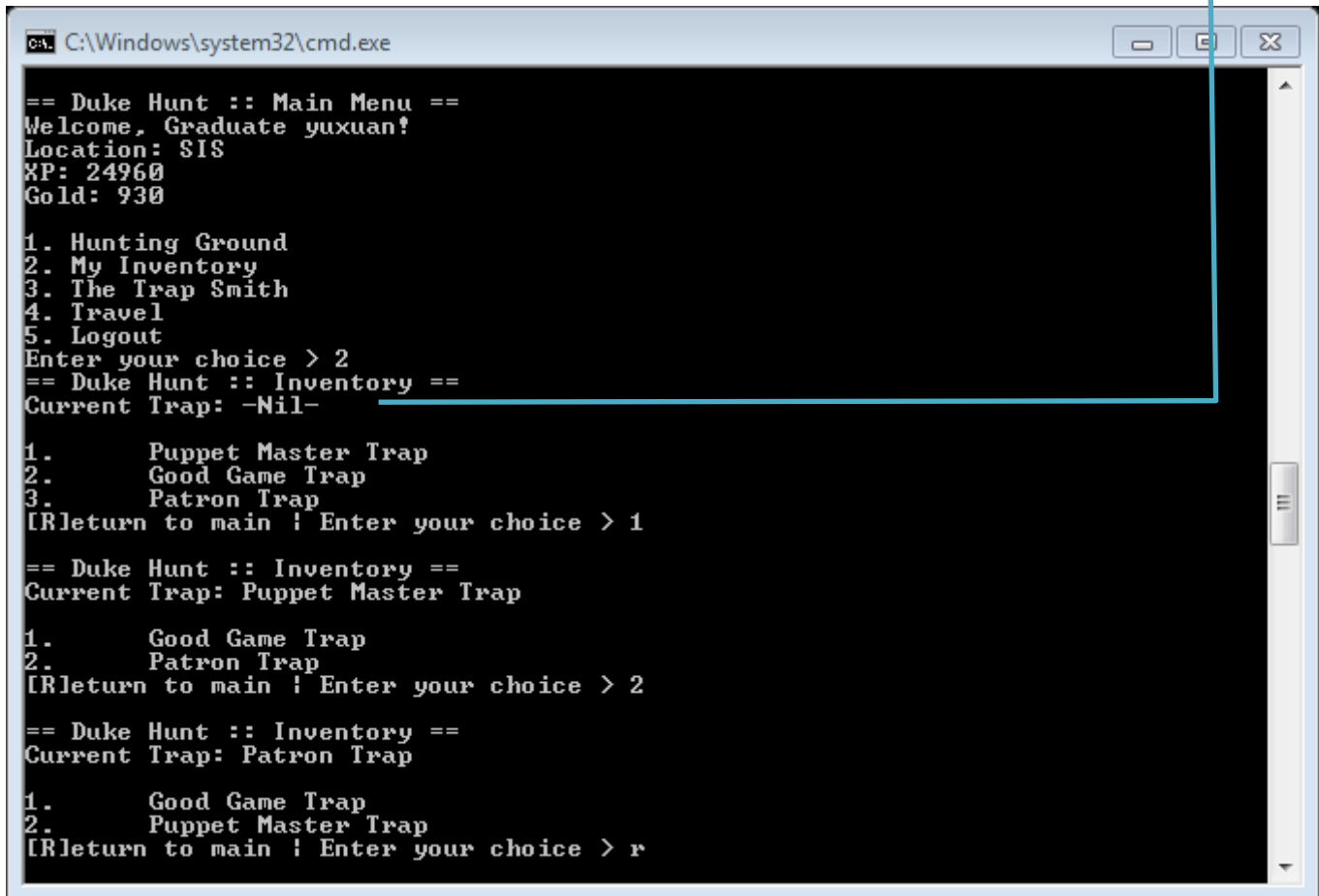
Oldest record
among the 10.

Newest record.

When the hunter sounds the horn, the newest record will be display at the top of hunting records, and the oldest record will be deleted in database.

Manage Inventory use case

The system will show “-Nil-” if the hunter has no equipped trap.



```
C:\Windows\system32\cmd.exe

== Duke Hunt :: Main Menu ==
Welcome, Graduate yuxuan!
Location: SIS
XP: 24960
Gold: 930

1. Hunting Ground
2. My Inventory
3. The Trap Smith
4. Travel
5. Logout
Enter your choice > 2
== Duke Hunt :: Inventory ==
Current Trap: -Nil-

1. Puppet Master Trap
2. Good Game Trap
3. Patron Trap
[R]eturn to main ! Enter your choice > 1

== Duke Hunt :: Inventory ==
Current Trap: Puppet Master Trap

1. Good Game Trap
2. Patron Trap
[R]eturn to main ! Enter your choice > 2

== Duke Hunt :: Inventory ==
Current Trap: Patron Trap

1. Good Game Trap
2. Puppet Master Trap
[R]eturn to main ! Enter your choice > r
```

Buy/Sell trap use case

If the hunter chooses to sell the last trap he or she owns, the system will show the invalid information.

C:\Windows\system32\cmd.exe

Welcome, Graduate chris!
Location: SIS
XP: 15325
Gold: 3060

1. Hunting Ground
2. My Inventory
3. The Trap Smith
4. Travel
5. Logout

Enter your choice > 3
== Duke Hunt :: TrapSmith ==
Your XP: 15325
Available Gold: 3060

1 Sell Trap of Least Resistance (+125 gold : 0 XP)
2 Buy Graffiti Trap (-2000 gold : 600 XP)
3 Buy Patron Trap (-2500 gold : 1400 XP)
[R]eturn to main : Enter your choice > 1
Sale of Trap of Least Resistance is rejected. Is this your last trap?

== Duke Hunt :: TrapSmith ==
Your XP: 15325
Available Gold: 3060

1 Sell Trap of Least Resistance (+125 gold : 0 XP)
2 Buy Graffiti Trap (-2000 gold : 600 XP)
3 Buy Patron Trap (-2500 gold : 1400 XP)
[R]eturn to main : Enter your choice > 2

== Duke Hunt :: TrapSmith ==
Your XP: 15325
Available Gold: 1060

1 Sell Trap of Least Resistance (+125 gold : 0 XP)
2 Sell Graffiti Trap (+500 gold : 600 XP)
3 Buy Patron Trap (-2500 gold : 1400 XP)
[R]eturn to main : Enter your choice > 1

== Duke Hunt :: TrapSmith ==
Your XP: 15325
Available Gold: 1185

1 Sell Graffiti Trap (+500 gold : 600 XP)
2 Buy Trap of Least Resistance (-500 gold : 0 XP)
3 Buy Patron Trap (-2500 gold : 1400 XP)
[R]eturn to main : Enter your choice > 3
You do not have 2500 gold and/or minimum 1400XP to buy Patron Trap.

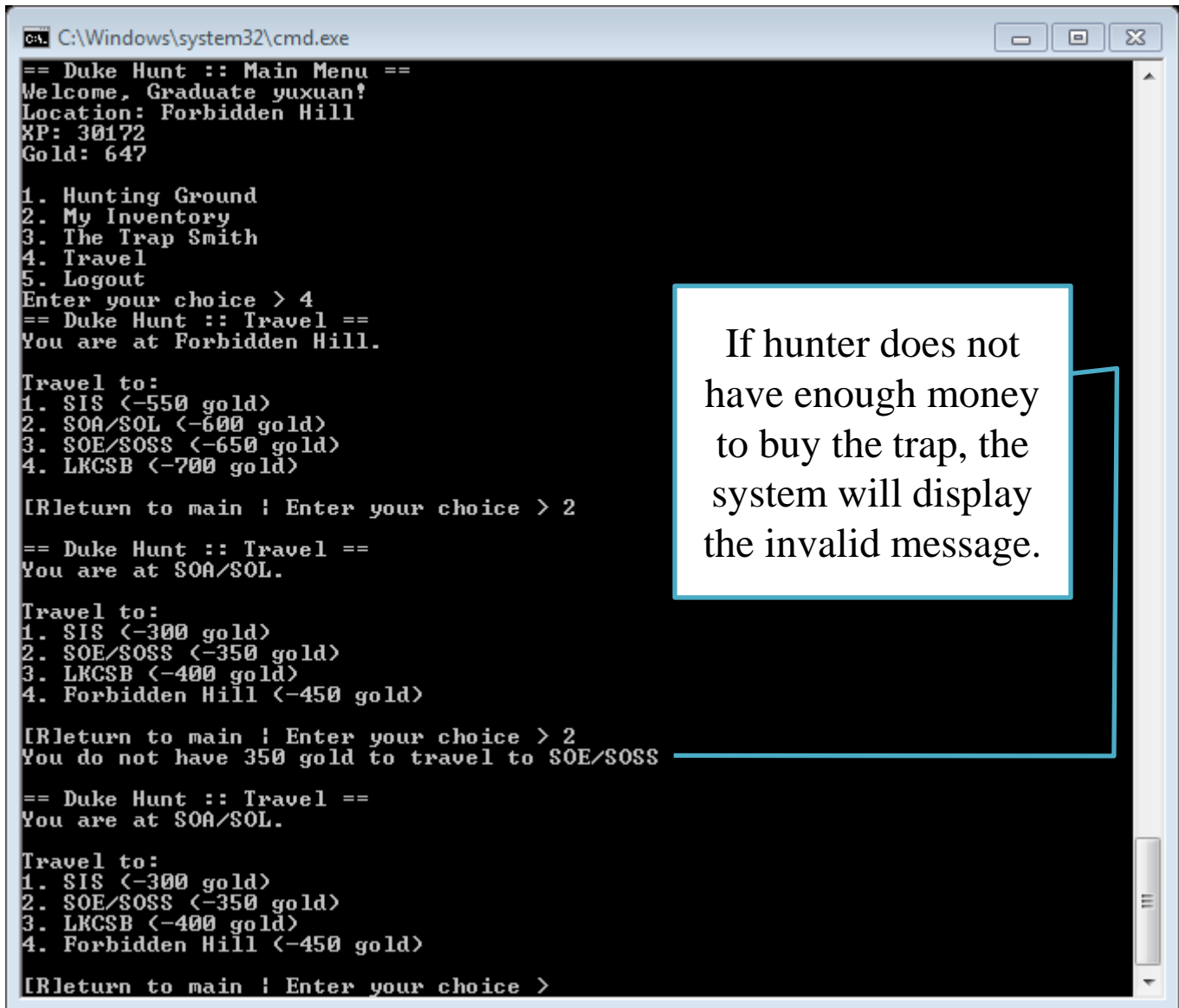
== Duke Hunt :: TrapSmith ==
Your XP: 15325
Available Gold: 1185

1 Sell Graffiti Trap (+500 gold : 600 XP)
2 Buy Trap of Least Resistance (-500 gold : 0 XP)
3 Buy Patron Trap (-2500 gold : 1400 XP)
[R]eturn to main : Enter your choice >

If hunter does not have enough money/xp to buy the trap, the system will display the invalid message.

Travel use case

G4-T03



```
C:\Windows\system32\cmd.exe

== Duke Hunt :: Main Menu ==
Welcome, Graduate yuxuan!
Location: Forbidden Hill
XP: 30172
Gold: 647

1. Hunting Ground
2. My Inventory
3. The Trap Smith
4. Travel
5. Logout
Enter your choice > 4
== Duke Hunt :: Travel ==
You are at Forbidden Hill.

Travel to:
1. SIS (-550 gold)
2. SOA/SOL (-600 gold)
3. SOE/SOSS (-650 gold)
4. LKCSB (-700 gold)

[Return to main ! Enter your choice > 2
== Duke Hunt :: Travel ==
You are at SOA/SOL.

Travel to:
1. SIS (-300 gold)
2. SOE/SOSS (-350 gold)
3. LKCSB (-400 gold)
4. Forbidden Hill (-450 gold)

[Return to main ! Enter your choice > 2
You do not have 350 gold to travel to SOE/SOSS

== Duke Hunt :: Travel ==
You are at SOA/SOL.

Travel to:
1. SIS (-300 gold)
2. SOE/SOSS (-350 gold)
3. LKCSB (-400 gold)
4. Forbidden Hill (-450 gold)

[Return to main ! Enter your choice >
```

If hunter does not have enough money to buy the trap, the system will display the invalid message.

Only region that is available for the hunter will display in the Travel menu.