Hangman Project

Hangman Final Project

Yenayet Edeglign Belachew

Yb222ce@student.lnu.se

Date 2022

Contents

1. Revision History

Date	Version	Description	Author
2022	**1.0**	**Project plan** **(**Vision, Project plan, Risk analysis, Strategies, Writing the skeleton of the Hangman game**)**	Yetnayet Belachew
2022	**1.0**	**Software Design** **(**UML diagrams, Modelling, playable**)**	Yetnayet Belachew
2022	**1.0**	**Software Testing** **(**Use-case, Unit test, Test Manual case, Report test**)**	Yetnayet Belachew
2022	**1.0**	**Final Project** **(**New feature, test final project**)**	Yetnayet Belachew

2. General Information

Project Summary	
Project Name	Project ID
Hangman Game	
Project Manager	Main Client
Key Stakeholders 1. **Players** 2. **Programmers/ Designers** 3. **Customers** 4. **Project Manager**	
Executive Summary	
Hangman is a guessing game which gives the players some attempts to guess the right letters to get the right words in the end if the player loses all his/her attempts he\she will be hanged. It could focus on marketing to investors and buyers. The game has rules and conditions. The game is played by one player and it is simple game. The game run smoothly with no bugs to attract the buyer and discovered what people liked about games and then made advertisements.	

3. Vision

The project aims to create and develop a Hangman game. Hangman game where a player can guess words. This game is common for most people and it comes to challenge our knowledge of choosing a word category. Most of us may know the principle of the Hangman game.

The game should be able to run and ask a player to start the game by provide a number to choose the level of the game and ask him, her to enter first characters however, the hanged man image will be displayed the body for each wrong attempt or guess. When the player starts playing the game will provide the player with three options one of them is to play an easy game by providing the player 7 attempts to guess the correct letter other, the second option by choosing medium game by providing the player 5 attempts for guessing the word and the third option by providing the player 3 attempts for guessing the correct word.

The words, of course, will be randomly from the list in the program. The concept of the game is when the player guesses the correct letter, the letter will be shown in the display like this ---b--- or if the letter is repeated it will display on the screen both repeated letters like this ---ee---. If the player guessed the correct word before he loses all his attempts player will win then survive otherwise the complete image of the hangman will be displayed, it means man dies and be hanged.

4. Project Plan

The project plan is to create and develop Hangman game. One of the plans is to make profits from this game and attract the users to play this game and discover what people like and put advertisement to get money from the game and sell it to the buyers. The game is creating also to give the players (children) more knowledge about words or name of countries or something else…

4.1 Introduction

Hangman is a game which will be created to give user opportunity to guess the right word by give the user some attempts, last if the user “player” guessed the right word, he\she wins. if he/she didn’t, he will lose and be hanged. This game will be updated through four iteration first iteration for planning the second on is modeling, third iteration is doing testing for the program and the last iteration is adding new features and make the game ready to play.

4.2 Justification

The main objective of creating the Hangman game is to divide the task into a smaller part that can be easier to handle it or to deal with. It could be also made to make children play and get more knowledge and teach them new

words like country names, food, animal and more things. It could be also made to get some benefits from it by putting advertisements on it.

4.3 Stakeholders

Players: who will play the game, or we can say the user who is given few chances to guess the correct word.
Developer/ Designers: People who designed the Hangman game, testing the code and add more features the game.
Customers: Those people which buy the product and get benefits from it like put advertisement on it or something else.
Project Manager: who is responsible on the plan of the project.

4.4 Resources

Software engineering book 10th Edition, Literature, Manpower, available time, and internet" oracle" …

4.5 Hard- and Software Requirements

Hardware: computer with basic requirements like IDE, which will be created by eclipse.

Software: Java, JDK, and Eclipse which we use it when we run the program which has JVM and run time, Eclipse program, Internet.

4.6 Overall Project Schedule

First assignment is: Documentation
Second assignment is: using UML for modelling
Third assignment is: using Junit Test
Fourth assignment is: to complete the final step on project

4.7 Scope, Constraints and Assumptions

Scope is the project main goal of implementation the Hangman game in java. It is playable game which mean that we are going to use Eclipse, so we are not going to design the game, it is just a console application.

This game only focuses on one player not more.

Constraints: It is the executable operation in a terminal or console and man should have a computer to run the game.
Assumption: It just if user know to deal with the game. User should know the basic of Hang Man game like to run it and understating how install it etc.…

My game is going to be simple and smoothly. My game also is going to give the player 7 attempts for the easy level to guess the correct letters, 5 attempts for the medium level and 3 attempts for the hard level and show him the right letters until player can get the correct word.

5. Iterations

In all 4 iterations are require something different from iteration to another iteration and all iterations are very important to each other as start from planning, documentations, code, adding or update the project and test it and make sure that all requirements are achieved.

5.1 Iteration 1

It is the first iteration in project Hangman it contains the project plan and how to create the project, which use to complete the documentation and implementation, so implementation goals are meet in code that I have already implemented to start working on it.

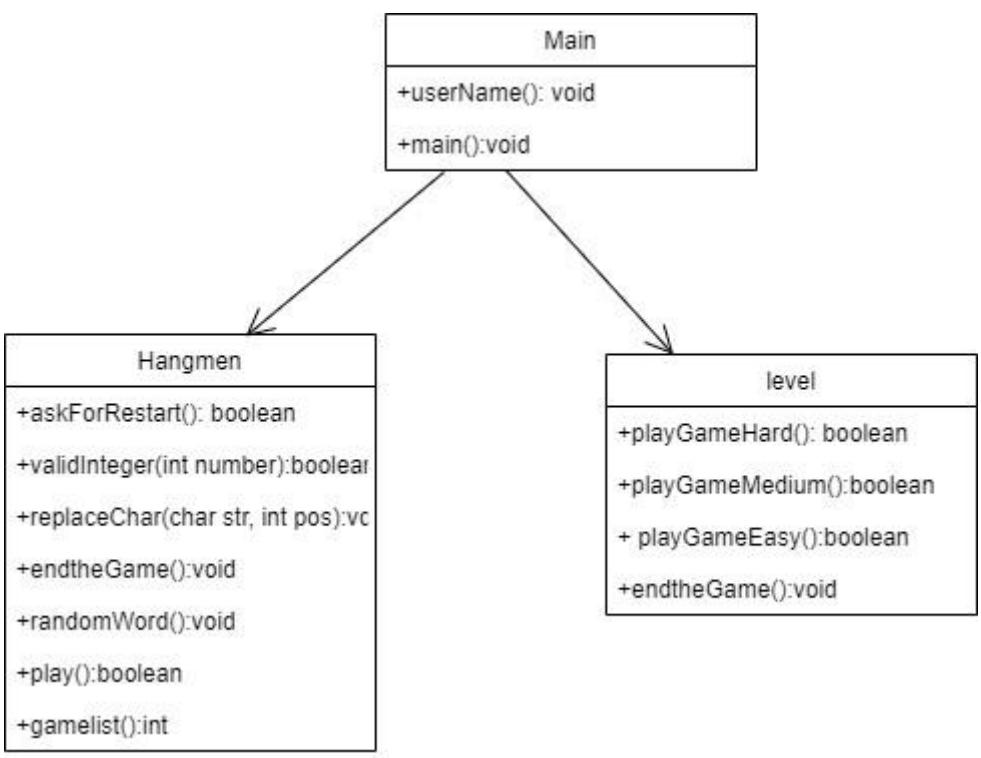
Reflection:

This is first software project for me was new and challenge for me, because that was something, I am not used to do it or practice it. I have learnt how to make a good plane before I start any project and divide the project to part until I can find the problem or the risk and fix it or terminate it, I also got more knowledge about documentations.

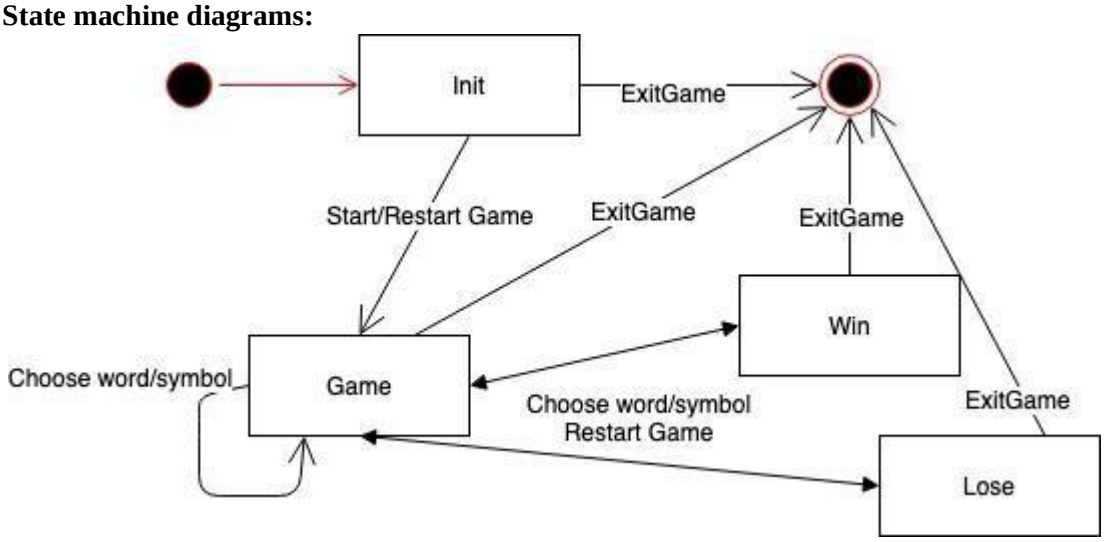
Subject	Estimated time	Real-time	Start Date	Finish Date
General Information	20 minutes	20 minutes	2020	2022
Vision	40 minutes	2 hours	2020	2022
Project plan	3 hours	5 hours	2020	2022
Iteration	1,5	2.5	2020	2022
Risk Analysis	50 minutes	50 minutes	2020	2022
Code	40 minutes	30 minutes	2020	2022

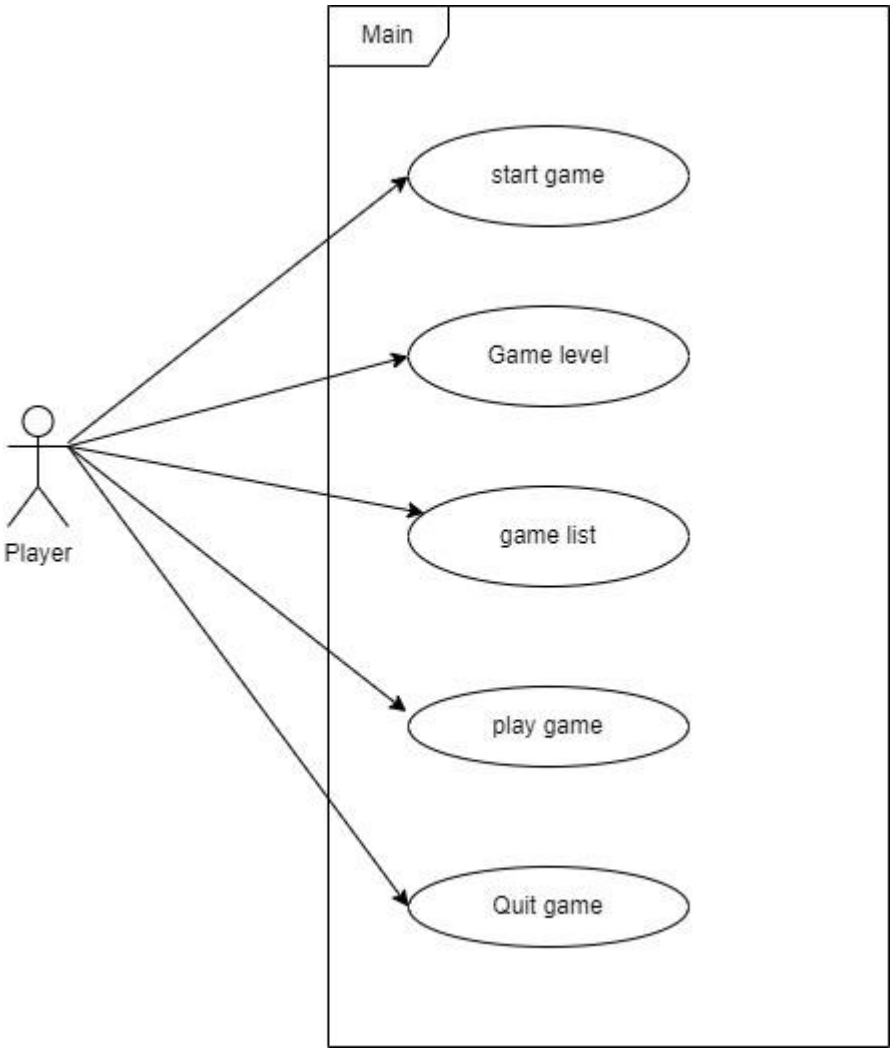
5.2 Iteration 2

The concept of this iteration is to a playable version of the Hangman game at the end of this iteration. I added some features and using UML class. As Usual game starts by asking user to provide his/ her name then after providing user name is to start the game then coming welcome message and ask the user to choose the level then choose name of categories “cities or cars name” to start guessing the unknow word, if the guessing word is correct system will provide winning message and ask user to play again or quit the game, but if user field system will display hangman and ask the user also to play again or to quit the game.



State machine diagrams:





Use case Diagram:

Use Case UC1	**Starting game**
Primary Actor	User plays the game
Stakeholders	Who wants to play a game session? Developer: who wants to provide the user with a good experience.
Main scenario:	1, The program will begin when the user has entered an input as a name. 2, The user enters his username. 3, The game will show the user a menu to choose from options. 4, The welcome message is played. 5, The user Chooses the level (see UC2 below).

Extensions	The player chooses to exit. Game will end
Precondition	None.
Post conditions	The game menu will display.

Use Case UC2	**Game level**
Primary Actor	User plays the game
Stakeholders	User: Who wants to play a game session. Developer: who wants to provide the user with a good experience.
Main scenario:	1. The user selects level, which may be suitable for him by click 1 to play easy, 2 to play medium, and 3 to play hard level. 2. The user will enter a number between 1 and 3. 3. Go to use case3
Extensions	Users will choose to quit the game. The game will end.
Precondition	UC1
Post conditions:	The player will get different attempts based on the chosen level.

Use Case UC3	**Game List**
Primary Actor	User plays the game
Stakeholders	User: Who wants to play a game session. Developer: who wants to provide the user with a good experience.

Main scenario:	1. The user selects which kind of categories of a word he/she wants to guess (3. Countries name, 4. Car brands). 2. Users choose 3 or 4 to continue the game. Go to UC4
Extensions	The player chooses to leave the game. The game will end.
Post conditions:	The user will guess a word based on the chosen category.

Use Case UC4	**Playing a game**
Primary Actor	User plays the game
Stakeholders	User: Who wants to play a game session. Developer: who wants to provide the user with a good experience.
Main scenario:	1. The user starts guessing and tries to guess the right letter to get the missing word. 2. Users will guess the right letters and win. 3. Go to UC5,
Extensions	1. the user starts guessing and tries to guess the right letter to get the missing word. 2. Users will run out of attempts and losses. 3. A hangman picture will be displayed. 4. Go to UC5.
Post conditions:	Results will be displayed.
precondition	UC3

Use Case UC5	**Quit the game**
Primary Actor	User plays the game
Stakeholders	User: Who wants to play a game session. Developer: who wants to provide the user with a good experience.

Main scenario:	1. The user will be directed to the final menu where the user has two options to choose from (1. Restart 2. Exit). 2. The user will Choose 2 to quit the game. Game ends.
Extensions	Users choose 1 to restart the game. Go to UC1
Post conditions:	The game will end.
precondition	UC4

Theme 2	Estimated time	Real time	Analysis
Revision History	10m	10m	
Iteration	15m	15m	
State-machine	1h	1h 15m	I was new for UML diagrams and Modelling.
Class diagram	30h	30h	
Use Case diagram	2h	2h	
Codes	13h	16h	Try to figure out how to start writing the code
Dress Use cases	2h	3h	Creating use cases for use case diagram
Total	18h 55m	22h 55m	

Reflection In this iteration 2 I have learned how to create state machine, class diagram, use case and dress use cases it was good practicing. I was really interesting by using UML class. I have completed the code for the project in that iteration

5.3 Iteration 3

It focused on testing the cases and code that I built and could be also extended from iteration 2. In this iteration it could focus also on “use cases” user’s information and test it and check it, if it is working (valid) or not.

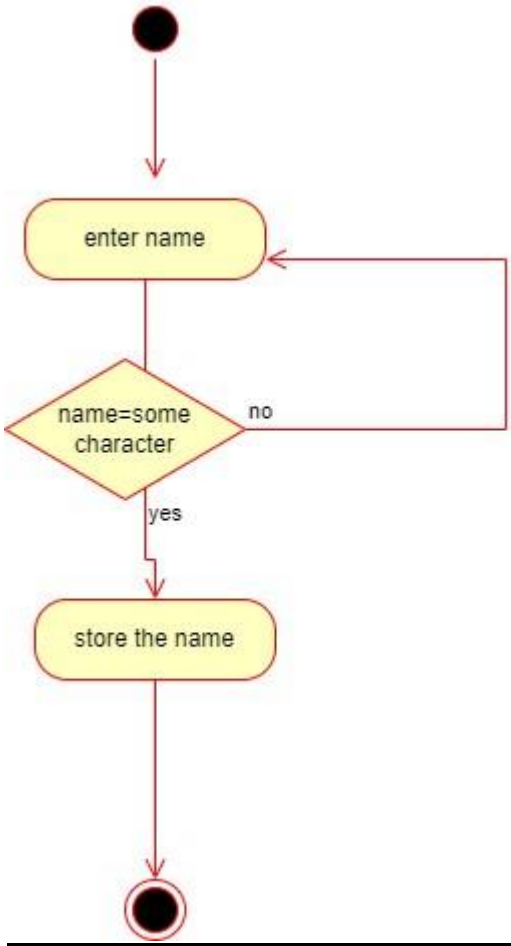
Link to my GitHub repo:

The game start with username, it requires the player to enter username to start playing so I can call this step:

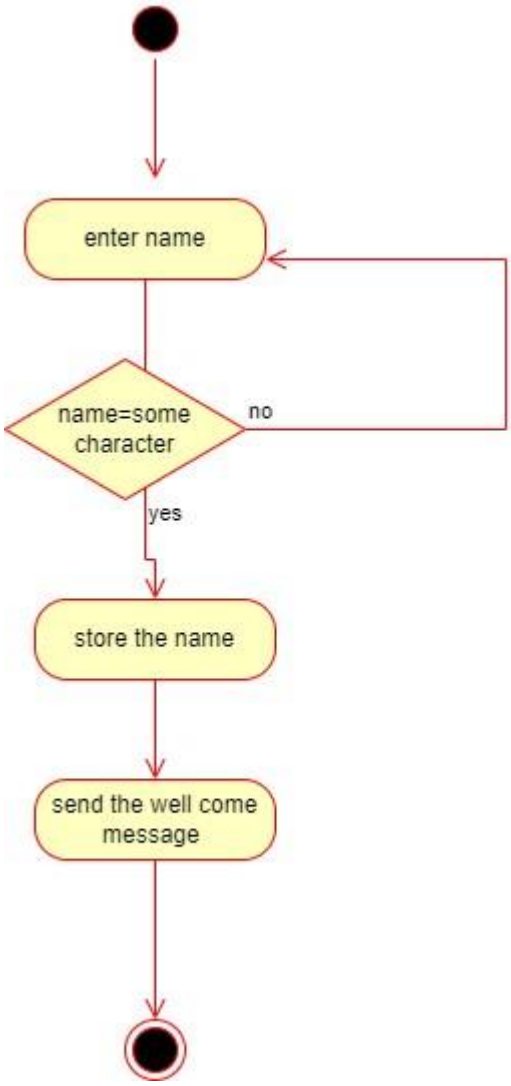
Use-Cases

The game start with username, it requires the player to enter username to start playing so I can call this step:

Use Case ID:	**UC-1**
Use Case Name:	Store username
Primary Actor:	user
Description:	User needs to write his\her name. Store to the system.
Post conditions:	a username is sorted
Basic Flow:	• Starts when a user wants to enter her\his username into the game • The system asks for a username • The User provides the username • The system stores the username and shows that it has been stored
Alternative Flows:	• the user provides an empty username • The system shows an error message “Enter your name again!”



Use Case ID:	**UC-2**
Use Case Name:	welcomed
Primary Actor:	system
Description:	Store the name and send message to user screen
Post conditions:	a user has been welcomed
Basic Flow:	• The user starts Hangman game • The system shows a welcome message including the username "Welcome to Hangman game!"
Alternative Flows:	• the user has not entered a username • Goto "UC1 Store a Username" • Goto Step 2. in the Main scenario



Test plan

The main objective in this iteration is to test part of the code that was implemented in iteration.
The manual test will be in a static manner by following a of scenarios and analyzing whether the output corresponds to the expect one or not. Unit Test will have a dynamic approach and is done by writing and running dynamic manual test cases. The reason for testing these use cases is to test the fundamental functionalities of the project.

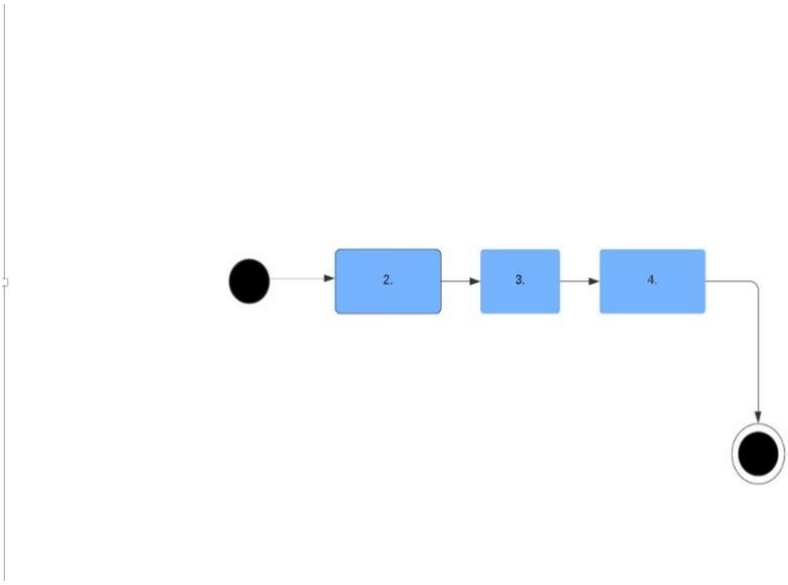
Project plan

Time plan **Theme 3**	**Actual**	**Estimated**
Test plan	20m	20m
Manual test cases	3.5h	4h
Unit tests	1h	1h
Code	16h	11h
Test report	2h30mh	3h
book	8h	9h
Documentation	31h 20m	35h
Theme 2	**Duration**	**Estimated**

Test Manual case

TC1.1 User valid input

a- I can start with correct Username scenario



Main scenario

(invalid username) it is teste the user input Pre-condition:

Start the game.

Steps:

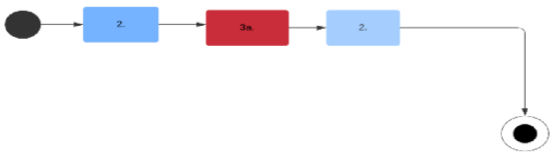
- Start the game
- Show a message to provide invalid name
- Enter username to start paly Expected
- Welcome message
- Show choose to play easy level by enter number 1 or medium level by enter number 2 or hard lever by enter number 3 or number 4 to quit the game.

TC1.2 user made an invalid input

Use-Cases: UC1 checks user's input

Scenarios: Invalid Username

This scenario forces the user when he input rather English alphabets or negative integers.



Steps:

- Start the game
- Show message to enter username
- Enter invalid or wrong username some contains not English letter or *, #System will show error message to write English letter or positive number System force user to write valid username and system waiting for the valid username to start the game.

Expected

- The system should display the error message "English Letters Or positive integers!"

- The system prompts the user to enter a valid name

Test report -TC 1.1, TC1.2

Test traceability matrix and success

Test	UC1	UC2
TC1.1	1/OK	0
TC1.2	1/OK	0
Coverage and success	2/OK	

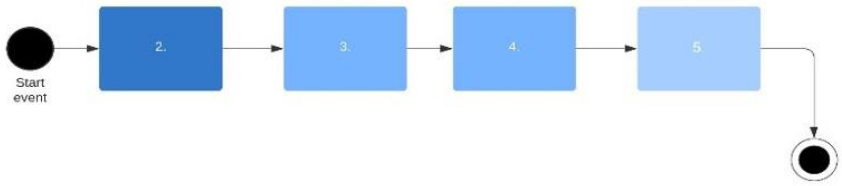
Automated unit test coverage and success

Test	Main	Hangman Test	**Hangman**
UserName_Test	0	0	100%OK
Coverage and success	0/NA	0/NA	100%OK

All the tests successfully passed, next iteration I need to focus on UC2, and the codes inspected, and most classes require mocks to be testable.

Use-cases: UC2 checks user's input Scenario:
checks user's input successful.

The main scenario of UC2 is tested where a user made a valid input to quit the game.



Steps for testing:

- Finished the game by know the unknown word
- The system greets the user upon wining
- System shows menu

1. Press "1" to restart and play again

2. Press "any positive integer" to quit the game except 1 because one is restarting the game

Enter "3" and press enter

Expected

• The system terminates the game with a message "Thank you and Goodbye"

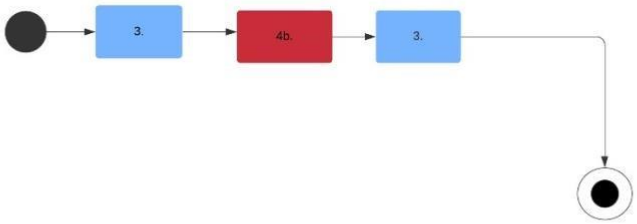
TC2.2 user made an invalid input to quit the game

Use cases: UC2 check user input to quit the game

Main **Scenario:**

The user enters a negative integer. Refer 4b under alternative scenario.

The alternative scenario comes into force while user inputs a negative integer or something not English letter and show an error message in the display



Pre-condition: The unknow word is shown in the display

Test steps

- Finished the game by shown the unknow word
- Show greets message by system.
- System shows menu

1. Press “1” to restart and play again

2. Press “any positive integer” to quit the game except 1 because one restart the game

• Enter “-3” and press enter

Expected

• The system should show an error message

• System displays “provide a positive integer”

• The system waits for the input

Test report TC 2.1, TC2.2

Test traceability matrix and success

Test	UC1	UC2
TC2.1	0	1/OK
TC2.2	0	1/OK
Coverage and success	0	2/OK

Automated unit test coverage and success

Test	UserName_Test	Main	**Hangman**
Hangman Test	0	0	100%OK
Coverage and success	0/NA	0/NA	100%OK

Unit Test

Source code 1:

This source code for username input, it requires valid username

```
public static void userName(){
    Scanner sc = new Scanner(System.in);
    String name;
    System.out.println("Enter a name to continue: (only English letters or integers)");
    name = sc.nextLine();
    while (!name.matches( regex: "[a-zA-Z0-9]+")){
        System.err.println("English Letters Or positive integers!");
        System.out.println("Enter a name to continue: (only English letters or integers)");
        name = sc.nextLine();
    }
    System.out.println(name + "! Welcome to the Hangman game");
```

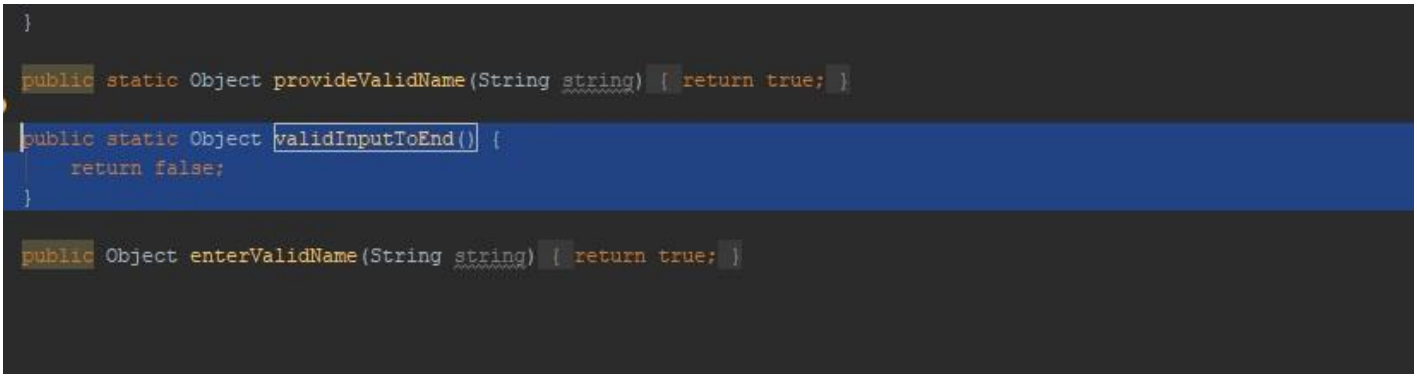
Junit & Result of source code 1:

```
@Test
void testEnterValidName_Valid() { // this is for right right input user name , english letter or integers
    assertEquals(true, man.enterValidName("talha"));
    assertEquals(true, man.enterValidName("jack"));
    assertEquals(true, man.enterValidName("mora"));
}

@Test void testEnterValidName_NotValid() {
    assertEquals(true, Hangman.provideValidName("0'*==/)"));
    assertEquals(true, Hangman.provideValidName("+4K"));
    assertEquals(true, Hangman.provideValidName("-.*"));
    assertEquals(true,Hangman.provideValidName("!#&.+"));
}
```

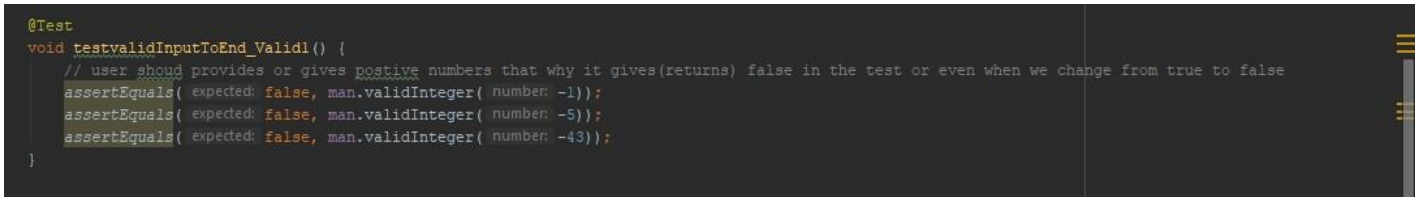
If user (player) enter valid user name, like English letter or integers numbers it will go ahead to start the game, but if user name entered invalid user name like *, # &!+-,,, or not integers or English letter, system will keep asking the user to enter the valid name.

Source code 2:



This is correct testing for one method, it does not give us a number less than 0 because input for user should be of course more than zero to start the game. Even if you focus you will see the method are true. In that place if we change true to false it will give one failure or error in the testing because it provides positive numbers and it says it is wrong, I'm going to give you another example for Junit.

Junit & Result for Source code 2:



We will see the first test numbers are positive and true it means this is true and with second test it gives negative numbers, but when users it says in the method is false it means this is also correct

Result of source code 2 It contains Error

It could happen when the user input wrong number to exist(quit) the game and this happen by enter wrong number or negative or even number which not following the structure game or rules. We will see the method it has true and this is false, that's why it gives error (bug) or failure in the test.

```
@Test void testEnterValidName_NotValid() {
    assertEquals( expected: true, Hangman.provideValidName( string: "0'*==/)"));
    assertEquals( expected: true, Hangman.provideValidName( string: "+4K"));
    assertEquals( expected: true, Hangman.provideValidName( string: "-.*"));
    assertEquals( expected: true,Hangman.provideValidName( string: "!#&.+"));
}
@Test
void testvalidInputToQuit_Bug() { // this method should return true to fail,  but here i wrote true to pass thats why gives correct tset.
    assertEquals( expected: false, Hangman.validInputToEnd()); // this is also test bug in the code,and test fail to quit the game,,

}
```

Reflection

In this iteration 3 I have used cases and a manual test cases, in these cases I clarified the user input if its valid or invalid and make the user free to choose or quite the game or continue. I improved my skills by writing cases and test it in this iteration, that was the most difficult iteration. I have learned how to write Junit test and more things as what scenario is and how to write it. This Iteration took time to do it because I did something new as write cases and main scenario and alternative scenario or writing Junit test.

Time log 3

Theme 3	Estimated Time	Real Time	Analysis
Extend and updating State Machine Diagram	30m	30m	I just added (completed) the diagram
Testing and updating the code	2h	3h	I got some problem with testing the code then I fixed it, that is why I took more than estimated time
Use cases 1 and 2	4h	3h	

Test Manual cases	3h	2h	
Time Log	20m	20m	

5.4 Iteration 4

This is the final iteration in the hangman game. I just added new level feature which is medium level. It is level between the easy and hard level and gives the player 5 attempts to guess the word. I added this new level also to Use case diagram and class diagram. last iteration is the outcome of this iteration is to complete the game.

Category	Estimated Time	Real Time	Analysis
Testing and updating the code	2h	3h	I got some problem with testing the code then I fixed it, that is why I took more than estimated time
Implementing new feature	4h	3h	Added a medium level to game menu
Debugging	3h	2h	Updated the previous iterations

6. Risk Analysis

6.1 List of risks

Most of project or program face risks with code itself or between units, by identify the risk it could be easier to reduce the risk overall the project.

We can say that risk is an expectation of loss information. Risk is something that you'd prefer not to have happen. Risks may threaten the project from hacking the software that is being developed.

There are some of risks we mush to focus on them**:**

-Lack of the new idea and experience.

-Losing data,

-late delivery for project short time to delivery or to update the project, -Unexpected risk like system failure

6.2 Strategies

For success any project, it should start with good strategies, a good plan must be created and divides the task into parts so that can discover or find if something wrong happened in the project or program and that make it easier to find the problems and fix it. We should think about some strategies like:

- Backup the system: this is the important operation to you keep a copy of system project and that step can keep you secure from the following list under:
- Protect the project
- System failure
- Viruses issue

6.3 Reflection

This is the last iteration I did not do many things here I just added new feature which is medium level which gives the user 5 attempt to guess the correct word. In this iteration 4 I put all iterations in one file to complete the project and to make the game is ready for user to play, so in the final I

mentioned some strategies and list of risk because that could happen when developer create the project, so the good Strategies come with a good experience and knowledge.

I think my project that I am working on it, it is not hard, it is easy to understand. I have learned also how to except the risk and be ready for it, and how to plan to face the risk by doing backup for the project and reduce the risk by doing risk analysis to make project success.

In the beginning of iteration 1 was confused because I did not get the concept or the point of this project, but after I started to figure out how to start this project. Seriously I am getting better now of course better than the beginning. Finally, I think in the real-life job we going to use that a lot, so that was only the beginning.

7. Time log

Iterations	Estimated time	Real time	Analysis
Iteration 1	2 days	3 days	First step in the game, plan, structure, and basic code
Iteration 2	3 days	3 days	Completed code has been added and use cases
Iteration 3	4 days	5 days	In this iteration have been created manual and automated JUnit test.
Iteration 4	3 days	3 days	Place all iterations in one place and added new feature which is medium level

Handing in

All assignments have some files to hand in. The overall advice is to *keep it simple*. Make it easy for the receiver to understand what the files are by using *descriptive* file names. Use as *few* separate documents as possible. Always provide a *context*, that does *not* send some diagrams in "graphics format", but always in a document where you provide the purpose and meaning of the diagrams. Remember that the "receiver" is a customer and as such has very little knowledge of the diagrams and documents – always provide context that makes anything you hand in understandable to a non-technical person.

To hand in an assignment, make a git release and hand in the link via Moodle to that release.