SOFTWARE DEVELOPMENT PROJECT HANGMAN

Assignment 1, 2nd attempt

RASHED QAZIZADA

Linnéuniversitetet

rq222ah@student.lnu.se

COURSE CODE: 1DV600

2019-04-13

1. Revision History

Date	Version	Description	Author
2019-04-12	1.0.2	Vision, Project plan, Writing the skeleton of the Hangman game	Rashed Qazizada
2019-04-12	2.0 .2	UML diagrams, Modelling, playable version of Hangman game	Rashed Qazizada
2019-04-05	3.0.2	Vision, Use-case, Test plan, Unit test, Reflection	Rashed Qazizada
2019-04-19	4.0.2	Plan, design and test for the final product	Rashed Qazizada

2. General Information

Project Summary	
Project Name	Project ID
Hangman	rq222ah_1DV600
Project Manager	Main Client
Rashed Qazizada	Hangman players (students)
Key Stakeholders	
• Customer • Project manager • End-users	
Executive Summary	
Hangman is a fun guessing game which can be played by one or more players and it is not age-restricted. The purpose of developing the Hangman game is that the player is going to guess a word by suggesting letter after letter. The player has limited number of chances/lives to guess and if the user guesses the correct letter the number of chances remains the same and for every incorrect guess the chances decrease by one. Finally, the user will be greeted upon winning or losing.	

3. Vision

Hangman is a game where you must guess words. Most of us may know the principle of this game. The game is to challenge your field of knowledge on the selected word category. In this project, Hangman game is going to be developed and the game should be able to run in the console window and will therefore be a text-based version where the hanged man image will be constructed for each wrong guess. However, before the user starts guessing the characters, they must provide a valid name then the player will be given extra option such as to show the high score board, play easy, hard and to quit the game.

Execution of the man in each round has 7 tries in easy level and 5 tries in hard level before the man is hanged. A word will be picked randomly from a predefined selected word category. List of nouns and the number of letters in the word will be represented as underscores. For instance, the word to be guessed is “Sweden” the word will be shown as _ _ _ _ _ _ if the player’s first guess is ‘e’ then it is displayed as _ _ e_ e _. Additionally, If the player manages to enter the right letters and wins 5 games in a row their name will be stored automatically on the high score board.

However, If the user guesses all the letters of the word correctly within the limited number of guesses, he wins the game but if the user fails to make the right guess till the end then the man dies and is shown by a complete image of a hanged man which indicates that player lost the game.

3.1 **Reflection**

First, writing a vision document helped me to understand the Hangman game project better and helped me to think different functionalities of the project Hangman game. As a beginner, I had no knowledge in regards how a project is done or how it is processed. Furthermore, writing this vision document helped to organize and limit my work on the project.

I think having a plan is always good to have. It provides an overview of the project to the targeted Stakeholders, and I found the vision important.

4. Project plan

The basic idea of project Hangman is to develop a text-based Hangman game where the player is going to guess a word by suggesting letter after letter in the console.

4.1 Introduction

Throughout the project, Hangman game will be developed through four iteration given below and are based on previous programming language. (Java).

Iteration 1 Process and Planning, iteration 2 Modelling and Designing, iteration 3 Use-cases, Test plan, Unit test and last iteration is the final product release.

4.2 Justification

The main objective of creating such an application is the formal functionality of the software and the practical use of documentation and planning. Documentation and planning would most probably keep programmers within the project scope. Therefore, is vital to divide a task into smaller tasks that can be dealt with step by step.

4.3 Stakeholders

customer: Plays the Hangman game designed by the programmer. The user is given a limited number of chances to win or lose the game.

Developer/project manager: Designs the game by constructing the code. Testing the functionality of the code and can add features like adding new players, removing a player, adding words to the word list.

End-user: The end user will come after using the product with some input in regard to their experiences.

4.4 Resources

The Hangman game has been created and developed in the latest Eclipse version i.e. JDK version 11.0.1 and also with the help of course materials.

Hard-and-software Requirements

1.1.1 Hardware

- A decent computer with basic requirements

1.1.2 Software

- JDK
- Eclipse
- Java

4.5 Overall Project Schedule

1. - **2019-04-19** Documentation, GitHub repository, skeleton code
2. - **2019-04-19** Assignment 2 UML behaviour- and structure modelling
3. - **2019-04-05** Testing Assignment 3
4. - **2019-04-19** complete the project final release. The project should be delivered to the customer.

4.6 Scope, Constraints and Assumptions

4.6.1 The scope: The project main goal is to implement the game “Hangman” in a text-based fashion in the language that I have used in previous courses (Java). The project will only focus on single player and the Hangman game is only playable using Eclipse.

4.6.2 Constraints: The main constraint is that it is only executable in a terminal or console and the user must have a computer installed with Eclipse, JDK, JRE and should know how to run the Hangman game.

4.6.3 Assumptions: The game will assume that the users understand the basics of Hangman or at least try to follow up the instruction given in each stage while playing the game. There will be a descriptive text with rules before the game starts. For example, there could be a line saying Enter a name or press 1 to play easy or similar.

4.7 Reflections

At the very beginning, I was confused with all the information, about how everything is going to work and step by step I figure it out. Scope, Constraints and assumptions gives a good overview the project. I think Project plan is beneficial and it makes the project clear for the next step. Despite the lack of required knowledge of documentation, I tried to follow the content of the template and complete the task given.

5. Iterations

5.1 Iteration 1:	The first iteration in this project plan alongside with some degree of implementation is to complete the documentation so that the implementation goals are met in code. I have already implemented an idea and skeleton code for my project to work with.
Tasks:	
Vision	• Create a vision document for the system and write reflection
Project plan	• Write a project plan for the project and write reflection
Risk Analysis:	• All projects face risks. During the risk analysis process consider each identified risk and make a judgment about the probability and seriousness of that risk
Coding	• Create the structure of your game. Push it to GitHub
	• Create project GitHub repository.
	• Pushing the project plan to the GitHub as a documentation file
	• Release Assignment one in GitHub. “the link”.
Time log:	• Create a time log containing the time estimate for each task
5.2 Iteration 2	The second iteration goal is to have a playable version of the Hangman game at the end of this iteration.
Tasks:	
ULM diagrams	• Get an overview of the Use Cases and the Actors (Primary and Supporting)
Play Game Use-case	• An option: to quit the game • An option: to go back to the menu when playing
Class diagram	• Creating a class diagram: each class should represent one file
Implementation	• playable version of Hangman game
Time log:	• Create a time log containing the time estimate for each task
5.3 Iteration 3	The third iteration focuses to perform tests on the system I built.
Tasks:	
Test plan	• What are the objectives of the testing? • What to test? • How this testing is going to be done • what should be dynamically or statically tested
Manual Test cases	• Name and ID of the test-case

	• Preconditions that need to be fulfilled • Test-steps • Check boxes if the test did succeed or fail
Reflection	• Write my experience and learnings
Unit Tests	• Create two automated unit-tests for each method in a total of four test methods
Time plan	• Create a time log containing the time estimate for each task
5.4 Iteration 4	The last part of the project is to complete the project. No new materials are added to the last iteration. The outcome of this iteration is to the complete the game
Tasks:	
	• The steps in iteration 1-3 must be reiterated for a set of new features • Plan, design and test for the final product
Documentation	• Final project document with complete instructions for how the game was built
Developer	• Must presents the project as a whole not individual

6. Risk Analysis

Indeed, in the process of developing a software you will face risks within the code itself or between the units that must interact inside the application. By identifying the risks, I as developer be able to proactively modify my codes and reduce the overall risk of my project. Risk is an expectation of loss a potential data that may or may not happen at the initial version.

However, when the risks identified, the strategies to modify the codes and ensure the project is a success

6.1 List of risks

- Lack of experience
- Lack of documentation.
- Lack of new ideas
- Lack of time, due to other courses at the university.
- Unexpected accidents like key students are getting sick at critical times in the project.

- The time required to develop the program is underestimated and failure to meet on agreed schedule
- Late delivery of the assignment
- Losing data, tool risk
- Technology, late delivery of hardware or software

6.2 Strategies

A good strategy begins by proper planning and documentation. In order to minimize the risk of mistakes at the very beginning of your project. Plan wisely

- Frequently backup the project.
- Consider extra time in the project.
- Finish the project before the delivery time.
- Find out bugs and other issues related to the software
- Plan more hours than usual but do not stress your staffs, this part needs to be carefully monitored to avoid exceeding the time and the budget.
- Respect and teamwork are keys to every project success, everyone has different skills. No one should expect all rewards and others should not feel their contribution to the project is undervalued

6.3 Reflection

I personally found this part of the assignment important and interesting. I think a good strategy comes with a good knowledge and experience. However, a project risks and strategy related to its size and coverage of the software and I believe the project I am expected to do is not a complex one but it was possible to come up with different strategies to prevent them and make the impact of them as little as possible. Although there might be more risks on the documentation as well as on the codes. I think, risk analysis is key to take the project to success, you plan ahead and find another way to tackle the problem.

7. Time log

Theme 1	Actual	Estimated	Reflection
Vision	2h	1h	Finding out what should be included in vision
Project plan	2h	2h	
Iterations	2h	2h	
Designing	1h	1h	
Create an account and a repository in GitHub	1h	3h	Never used GitHub and needed to go through instruction videos
Skeleton code	0h	40	I developed the Hangman game to a playable version for assignment 2 second attempt so at the beginning of the project I estimated 40 hours
Reading the book/lecture notes	8h	4h	Underestimated the length of videos and my learning capacity
Risk analysis	3h	2h	Read the book and went through different strategies to prevent risks