# HANGMAN PROJECT

**Osama Zarraa**
Linneuniversitet

Date
2019-02-08

# Contents


**1 Revision History** **2**

**2 General Information** **3**

**3 Vision** **4**

**4 Project Plan** **5**
4.1 Introduction . . . . . 5
4.2 Justification . . . . . 5
4.3 Stakeholders . . . . . 5
4.4 Resources . . . . . 5
4.5 Hard- and Software Requirements . . . . . 5
4.6 Overall Project Schedule . . . . . 5
4.7 Scope, Constraints and Assumptions . . . . . 5

**5 Iterations** **6**
5.1 Iteration 1 . . . . . 6
5.2 Iteration 2 . . . . . 6
5.3 Iteration 3 . . . . . 6
5.4 Iteration 4 . . . . . 6

**6 Risk Analysis** **7**
6.1 List of risks . . . . . 7
6.2 Strategies . . . . . 7

**7 Time log** **8**

**8 Handing in** **9**

# 1 | Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 2018-02-08 | 1.0.0 | Planning, writing the skeleton of the program | Osama Zarraa |
| | | | |
| | | | |
| | | | |

# 2 | General Information

| Project Summary | |
|---|---|
| Project Name | Project ID |
| Hangman game | 1DV_10001_HM |
| Project Manager | Main Client |
| Osama Zarraa | +13 age user |
| Key Stakeholders | |
| End users, developers, testers, teachers | |
| Executive Summary | |
| Hangman game is a word-guessing game with limited number of wrong guesses with extra features like user registration and statistics with points system to players. | |

# 3 | Vision

Hangman project is a text-based and a word-guessing game which consists of a word, players and an image that is drawn based on player's answer that is given character by character. Game starts with loading old files that represents players, scores and old games. After loading, a menu will appear telling the user to choose an option from:

- Registering a player
- Selecting an old player
- Start a game
- Showing players scores
- Exit

After starting a game, a "store" will choose a random word from a list which is loaded previously. Program will print an underscore characters represents the missed word, without showing the real characters, and players can guess the word by giving a character. Each time a player gives a character, program will check if this character is part of the missed word. If character is included in the word the program will replace underscores that matches the given character with the right characters. Otherwise a figure of a hangman will be completed by every wrong answer given by the players. The game has maximum eight or ten wrong answers to complete the hangman figure. If the figure of the hangman is completed the game will be ended and saved in a store for later statistics, else the player who gave the last successful character will be considered as a winner and a log will be saved as part of a point system. The program will also have a time limit which represents that the deadline of the game.

**Reflection**:

Writing a vision is an important step that gives a guarantee that everyone involving in this project shares the same vision. It is like giving the whole picture to everyone who will develops, tests or uses the result of the project. It also shed a light on the important steps of how the program works and what steps are required to make the program works in the context of the main client needs. Above this, it gives the features that are required in this program and who and where to implement them.

# 4 | Project Plan

I will start by defining the roles. Starting by myself as a developer and my teachers as a project managers and testers. My friend as an end-user.
After defining the roles, I will check the prerequisites such as IDE and a computer with java development kit for developing. Reading documentation for this project is also needed to accomplish this project.

Defining the roles, determining the prerequisites and reading the documentation will take about 8 hours. And It should be done by the end of 17th of January. After this step, we will enter the planning stage of the project and this step will end on 8th February 2019. Planning will include all case scenarios that are defined by the stakeholders with all features required to match the client needs. Planning stage will start with defining strategies and technologies needed to accomplish this project like programming language and libraries that will be used to accomplish this task and the learning curve that is needed to be able to use these technologies. This step will take 2 days and ends by 19th of January. After this step, we will determine the all use-cases needed in this project and the models needed in this project. After knowing all user-cases, I will start making the skeleton of the project by implementing the base classes for the player and the word. I will create the main class for the project and implement an xml-based store for saving and loading data from it. The deadline of the implementation of the skeleton is 8th February 2019.

After the planning phase, I will move to the next phase which is the full implementation of the game with making the state machine diagram for "Play Game". The deadline of this task is 21th February 2019.

The last stage of this project is the testing stage. This includes implementation of the testing methods and verifying that the final product meets the specified requirements. This also includes reading how the testing is done using Junit library and other libraries. The deadline of this stage is 8th Marsh 2019.

**Reflection**:
Project plan will define the major lines of the project with timeframes and all steps needed to accomplish this project. It also considered as a compass that will guide the programmer throw different steps of the project. Every developer needs some kind of project plan to follow and to prevent going out of the scope of the project.

## 4.1 Introduction

The goal of this project is to create a text-based hangman game with java within the timeframe of this course.

## 4.2 Justification

It is a requirement in Software technology course and an example of plan-driven project.

## 4.3 Stakeholders

End-users: persons who will use the final product of the project.
Developer: me as a developer.
Teachers: as a project manager.

### 4.4 Resources

Wikipedia, Software technology book, Java documentation.

### 4.5 Hard- and Software Requirements

Hardware: Computer with operating system such as Windows, macOS or Linux.
Software: IDE(Intellij), JDK (Java development kit), JRE (Java runtime environment)

### 4.6 Overall Project Schedule

8th February 2019: Project plan.

21th February 2019: Working version of the game with UML.

8th March 2019: Testing.

### 4.7 Scope, Constraints and Assumptions

Scope: Text-based game that is playable within the java IDE without a user interface. The console is the context of this game. This means that web-browser is out of our scope.
Constraints: Time is short for the whole tasks in this project.
Assumptions: Knowledge of Java IDE with JDK and java programming language to run the project.

# 5 | Iterations

## 5.1 Iteration 1

Writing the project plan with program skeleton that represented in the project plan. 8st February 2019 is the deadline of this iteration. Reading the resources from the book and checking the documentation about the game is also part of this iteration.

| ID | Description | Estimated Time(hours) | Actual Time (hours) | Deadline |
|---|---|---|---|---|
| D1 | Reading the documentation and resources | 16 | 24 | 8th February 2019 |
| D2 | Project skeleton | 8 | 12 | 8th February 2019 |
| Total Time | | 16 | 36 | |

## 5.2 Iteration 2

| ID | Description | Estimated Time(hours) | Actual Time (hours) | Deadline |
|---|---|---|---|---|
| D1 | UML Diagrams | 8 | | 21th February 2019 |
| Total Time | | | | |

## 5.3 Iteration 3

| ID | Description | Estimated Time(hours) | Actual Time (hours) | Deadline |
|---|---|---|---|---|
| D1 | Adding features | | | 8th March 2019 |
| D2 | Testing | | | 8th March 2019 |
| Total Time | | | | |

## 5.4 Iteration 4

| ID | Description | Estimated Time(hours) | Actual Time (hours) | Deadline |
|---|---|---|---|---|
| D1 | Completing the game | | | 8th March 2019 |
| Total Time | | | | |

# 6 | Risk Analysis

In this project the main three risks are: sickness, hardware failure and software crashing.

**Reflection**:
Being prepared for the risks is something mandatorily in SDLC (Software development life cycle). It is a way to keep the developer within the timeframe of the project and give him a chance to overcome them without affecting the whole project. Moreover, it gives the developer a future perspective of what would happen and how to make a backup plan for issues that will come.

## 6.1 List of risks

| ID | Description | Probability | Impact | Strategy | Solution if Occurred |
|---|---|---|---|---|---|
| R1 | Getting sick | 4 | 5 | Good planning | Get help from my colleges |
| R2 | Hardware failure | 3 | 5 | Using version control system like GitHub | Make a backup every time |
| R3 | Software crash | 4 | 5 | Frequently saving the files | Make copies of the project |

## 6.2 Strategies

For sickness issue, I will try to get help from my colleges and I will consider adding more time from the beginning to solve this issue.
For hardware failure issue, I will use version control system such as GitHub for preventing losing the project files.
For the Software crash, I will make copies of the files that I work on so I can go back to them if needed.

# 7 | Time log

| ID | Task | Estimated Time(hours) | Actual Time (hours) | Analysis |
|---|---|---|---|---|
| D1 | Planning | 8 | 8 | Creating the planning |
| D2 | GitHub Account | 1 | 2 | Creating and managing GitHub Account |
| D3 | Program skeleton | 8 | 16 | Creating the main classes of the program |