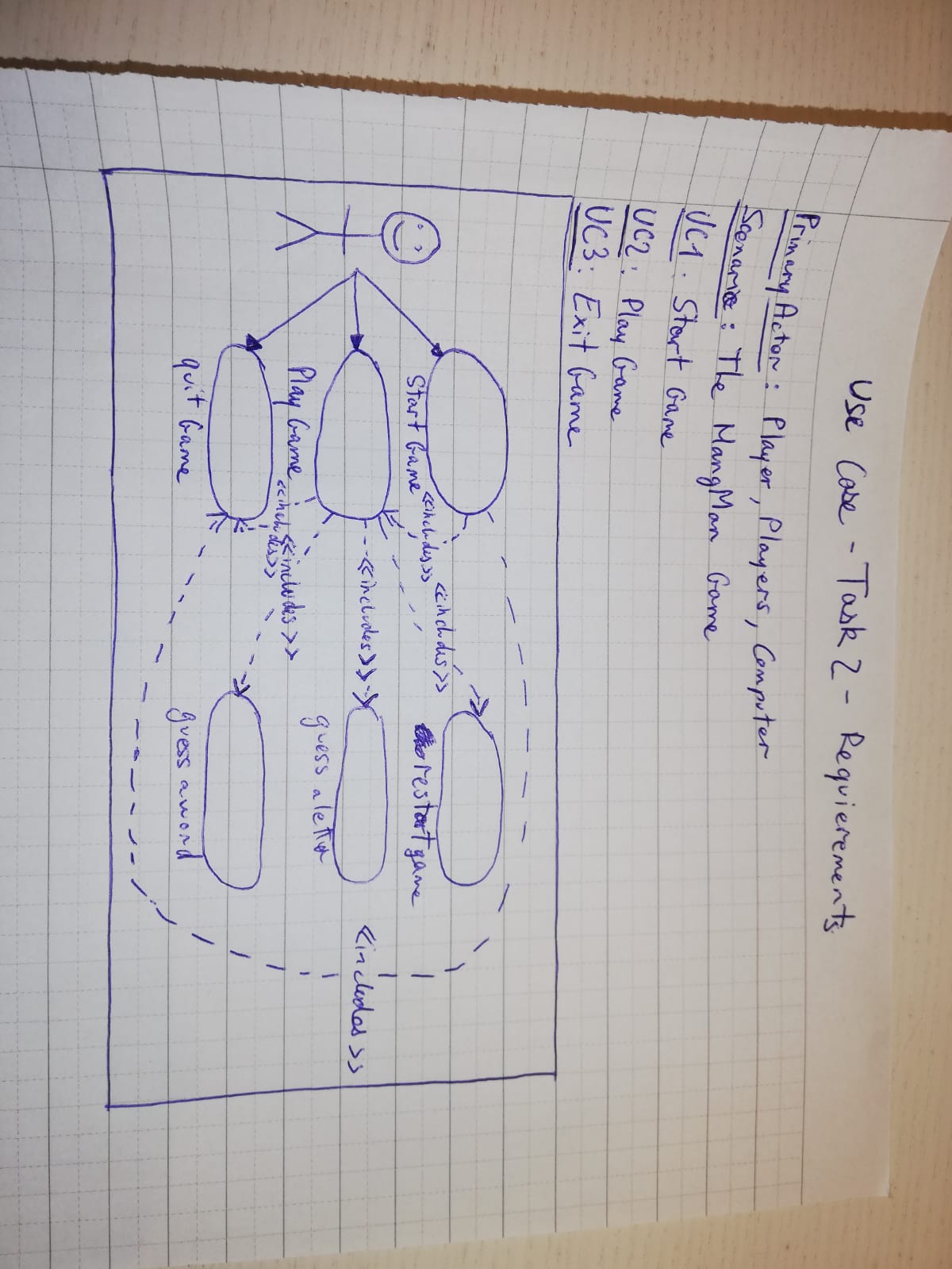
**Use Case, State Machine, Class Diagram**

**Task 1 – Time Log**

|  |  |  |
| --- | --- | --- |
| **Task to Do** | **Time Estimated (min)** | **Time Taken (min)** |
| Create Class Diagram | 15 | 20 |
| Write Fully Dressed Use Case | 30 | 40 |
| Create Use Case Diagram | 15 | 15 |
| Create State Machine | 30 | 60 |
| Implement Code | 60 | 90 |

**Task 2 – Requirement – Use Case diagram**



**Task 2.2 – Fully Dressed Use Case**

Precondition: The player clicks on the play button

Postcondition: Game is reset, and player plays again.

Main Scenario:

1. Starts when the player wants to play.
2. The system generates the secret word.
3. The player guesses a letter.
4. The system checks if the letter is in the secret word and update the screen with the result.

*Repeat from step 3 until the player has found the secret word.*

1. User selects between playing again or exit the game.
2. The system runs the program according to the player choice.

Alternative Scenario:

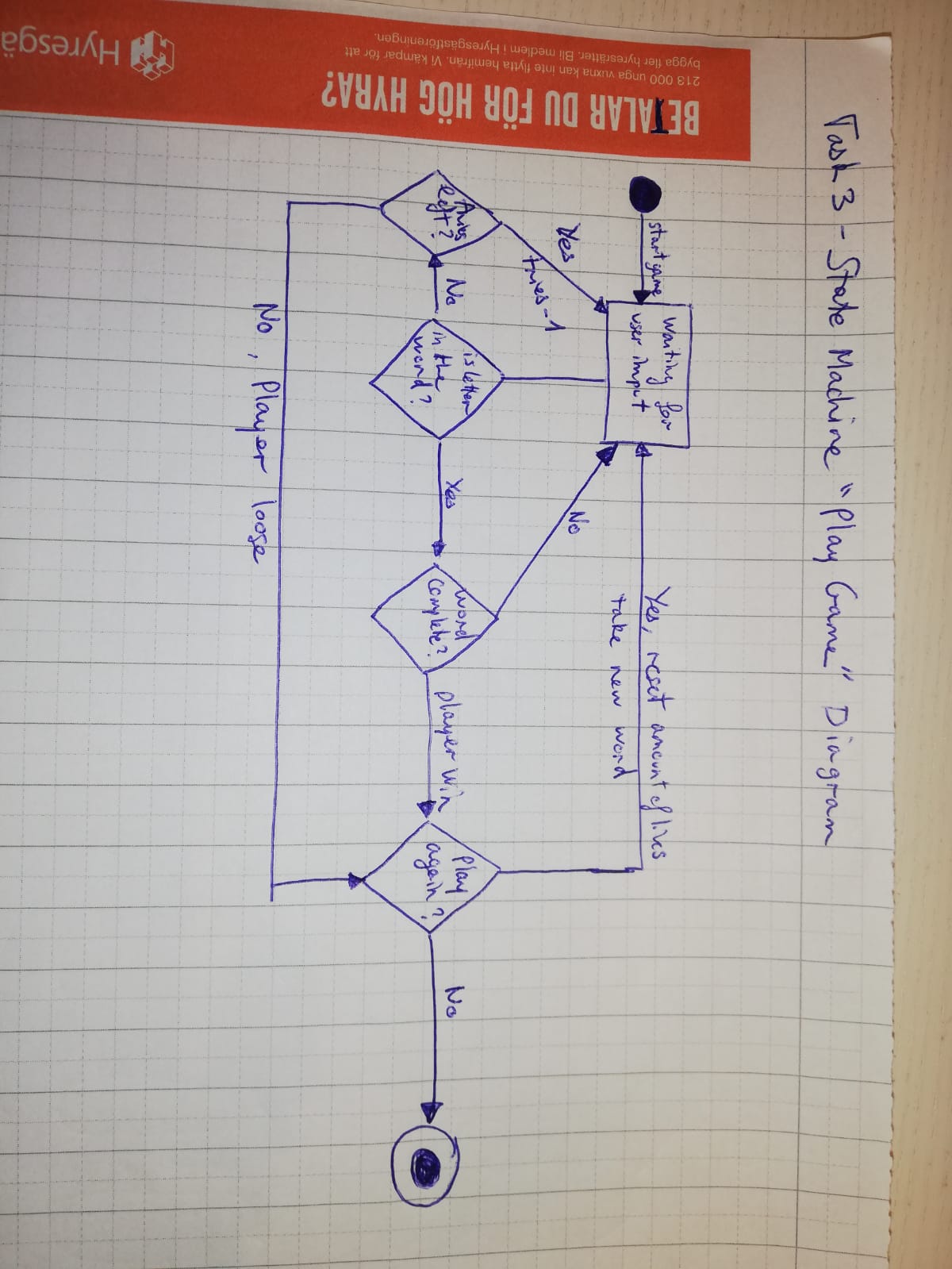
3.1 The player chooses a wrong letter.

1. The system reduces by 1 the amount of tries left and Go to 3.

5.1 The player makes the choice to exit the game.

1. The system exit the game (see UC3).

**Task 3 – State Machine of “Play Game”.**



**Task 4 – Modelling Structure**

|  |
| --- |
| **Hangman** |
| - **AllTheWords**: ArrayList<String>  - **WrongLetterList**: ArrayList<String>  - **Path**: String  - **DataBase**: File  - **Wordsize**: int  - **count**: int  - **TheWord**: String  - **WordArray**: String[]  - **UnderScoreA**: String[]  - **HMImages**: Image[]  - **Alphabet**: List<String> |
| + **getrandomWord**(): void  + **getWordArray**(): String[]  + **getCount**(): int  + **createUnderScoreArray**(length: int): String[]  + **getStringRepresentation**(UnderScoreArray: String[]): String  + **isLetterinWord**(guess: String, pos: int, word: String[]): Boolean  + **getWordSize**(): int  + **getUnderscoreArray**(): String[]  + **getWord**(): String  + **isEqual**(): Boolean  + **getWrongLetter**(): String  + **changeUnderscoreArray**(pos: int, change: String): void  + **main**(args: String[]): void  + **start**(primaryStage: Stage): void  - **CreateImages**(): void  - **getImage**(pos: int): Image  - **getAllTheWord**(file: File): void  - **changeWrongLetters**(WLetter: String): void  - **createWrongLetterLists**(): void |