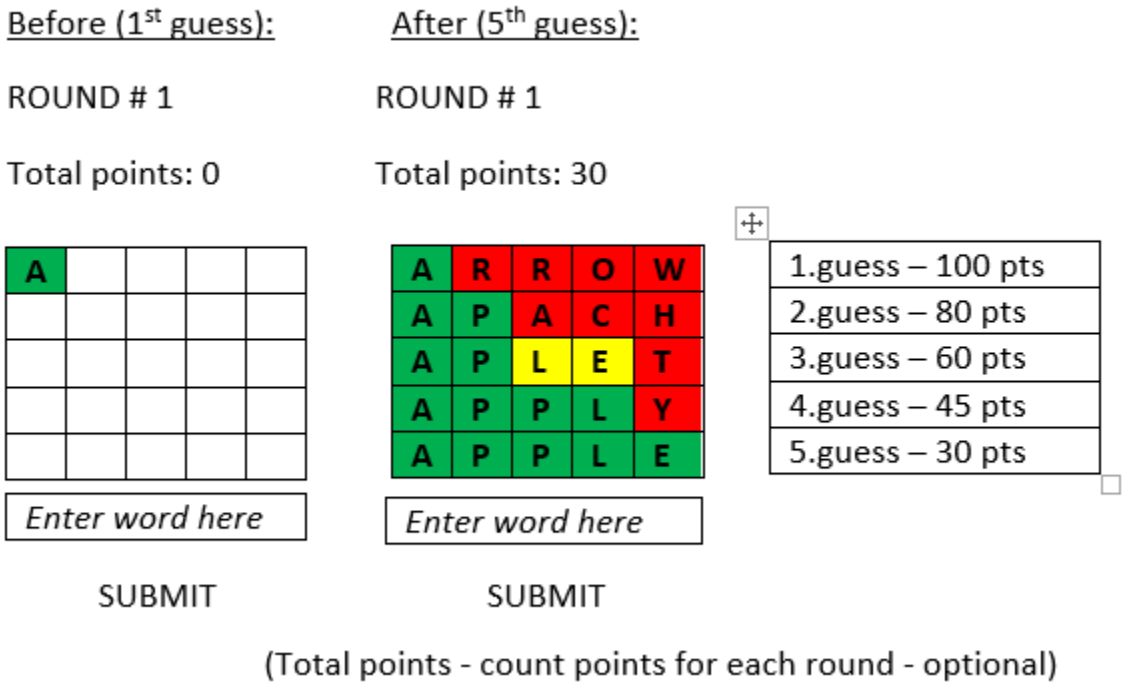
Contents

1. **Function requirements and game rules**

Game objective – guess the hidden word to win the game.

Game rules:

1. Player have 5 attempts to guess the hidden word,
2. First letter of the word is revealed,
3. Each letter's field of the entered word is highlighted on the board on a specific color:
    a. green color - word contains the letter and is in the correct position,
    b. yellow color - word contains the letter, but not in correct position,
    c. red color - the letter doesn't exist in the word.
4. If player guesses incorrectly 5 times, then game is ended.
5. If player guess the name within 5 attempts, then starts next round with new word.
6. To make game more interesting there are total points earned during the game. In each round player can earn points and the earlier player guesses the name more points player can gather: 1st guess – 100 points, 2nd guess – 80 points, 3rd guess – 60 points, 4rd guess – 45 points, 5th guess – 30 points. Highest score is saved for game.



Functional requirements:

1. Application "START GAME" window – **start button**
2. For the game is used **board** with 5 x 5 squares:
    a. 1st row (with 5 squares) displays the first guess, 2nd row displays the second guess, etc.
    b. At the start of the game first letter of **random word** in the first row is revealed in green color,
3. For the game **entry field** for the player guess is necessary:
    a. entry field length needs to be 5 positions,
    b. entry field allow only capital letters,
    c. in entry field can type only Latin letters (not symbols or digits) and if
        i. input is incorrect player is not allowed to press **submit button**,
        ii. input is correct player can submit entry.
4. When entry is submitted – entered word is displayed on the first row of the board according to the colors described in game rules for each letter:
    a. If a guess is correct - player earns 100 points that are added to **"Total points"** and starts new **round** with new word,
    b. If a guess is incorrect, player can make a new entry with the 2nd guess that is displayed in 2nd row in appropriate colors:
        i. if guess is correct – player earns 80 points and starts new round,
        ii. if incorrect – player can make a guess 3rd time (like on 2nd guess) and it can continue like this till 5th guess.
    c. If the player makes 5th guess and it is incorrect the game ends and "Total points" are displayed, but if the 5th guess is right – points are added and starts new round.
5. When gam is ends "Total points" are compared with **"Highscore"** and if larger, then new "Highscore" is saved until application is restarted.
6. Application "END GAME" window – **restart button.**

Summarization of requirements:

1. Application START GAME window, "IN GAME" window, "END GAME" window.

2. Game board 5 x 5 square
3. Random word– generated from database
4. Input field – for word entry (allow only capital letters)
5. Start button
6. End button
7. Submit button
8. Round Nr field
9. Highscore field
10. Total points field

2. System requirements

Hardware requirement

Standard operating system for the project is lightweight Linux distribution called Xubuntu, which is based on Ubuntu. It can be run as a native operating or as a virtual machine in VirtualBox. Minimum hardware requirements for the computer are following:

Parameter	For native system	For virtual machine
CPU	<15 years old	< 7 years old
RAM	>= 1GiB	>= 2GiB free RAM
HDD	>= 20GB	>= 20GB free space
Other	For teleconference: wired/wireless network card, sound card with microphone, permanent connection to the internet. Headphones are recommended.	

Operating System requirements

To ensure that all project members can perform tasks in the most efficient, **Xubuntu 20.04** should be installed as native operating system on the computer or in virtual machine. OS Set-up guide.

IDE requirements

To ensure that all project members can share the same configuration and work in a more consistent environment, **Eclipse IDE** **Version: 2020-06 (4.16.0)** should be

installed and used as the primary Java IDE on native operating system or on virtual machine.

Java requirements
In our case required Java version to develop code in is Java 11, this is due to the fact that Java 11 is latest version that offers Long-Term Support, by Oracle Corporation.

Spring framework requirements
Java 8 or 9 and Spring Framework 5.0.13. RELEASE or above. Maven version 3.2+, Gradle 4.x. Browser: Chrome / Edge / Firefox / Safari / Opera.

3. Market research.

Product: Word game Lingo (short, easy concept)
Target audience: female 25 – 50
Layout/design/accessibility: easy, simple, aesthetic
What are the main reasons for player to play?

- Learn new words
- Prove themselves
- Achieve goals
- Spend free time

What game needs to give to player?

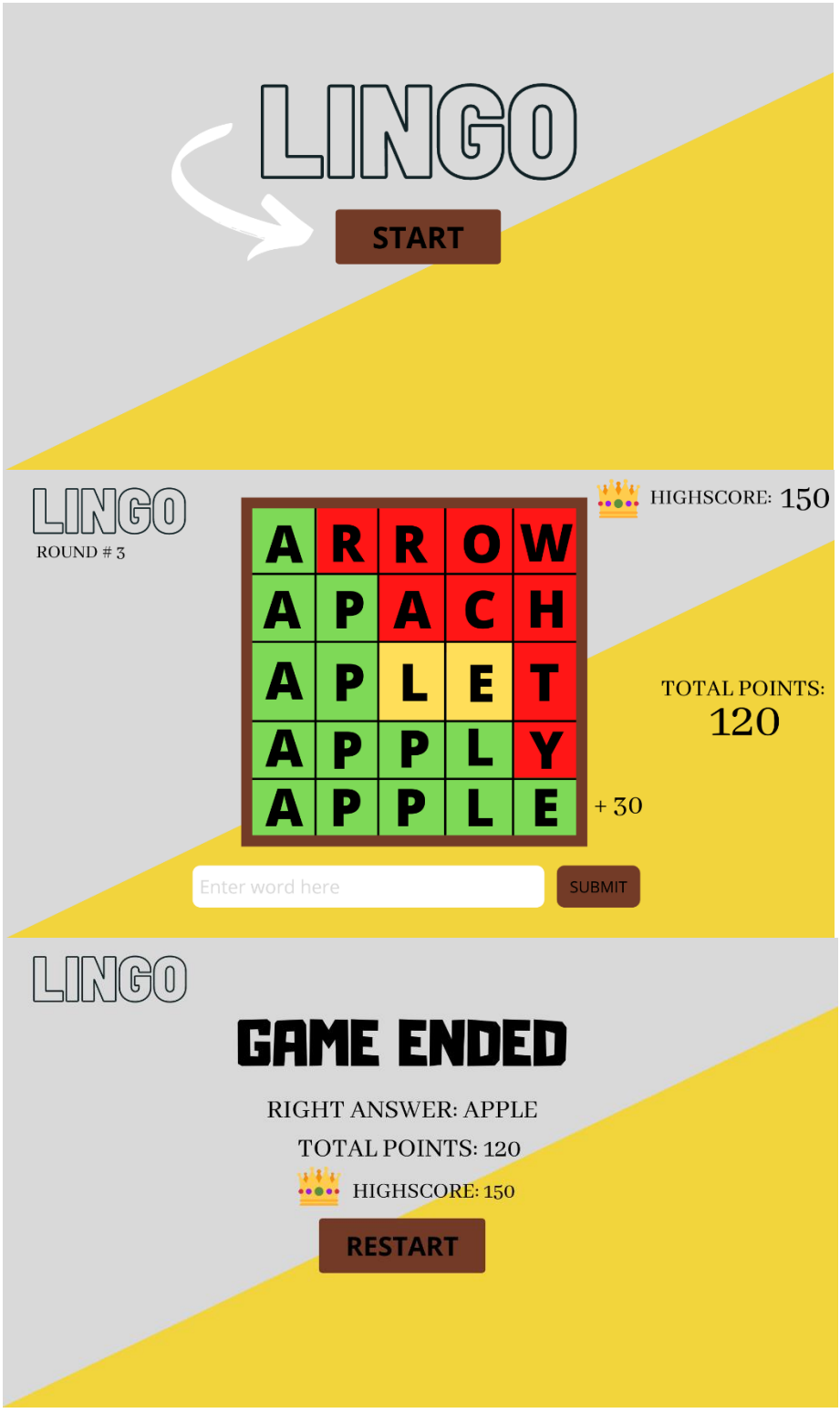
- Satisfaction
- Challenge
- Knowledge

Solutions:

- Words are divided into several levels- as you get more points words get harder
- Points are counted until player loses the game
- Best score is stored until page is reloaded
- Level system- when player achieves certain number of points, he/she reaches new level
- Suitable page design- simple, dark mode possibility

4. Design

4.1. BLUEPRINT interface (graphical design)



4.2. FRONT END design for application:

4.2.1. User interface layout:

- Game window fixed in the middle of WEB page
- Responsive WEB Design
- Application “START GAME” window – start button and headline the name of application “LINGO”
- Application “IN GAME” window – displays game board, entry field, submit button, Total points field, Highscore field, Round number field,
- Application “END GAME” window - displays “Total Points”, “Highscore”, headline “Game ended”, Restart button.
- Color scheme

4.2.2. Things to DO at development FRONT END stage:

- FRAME & Color sheme – game windows of start, in and end game stages.
- BUTTONS – start, submit, reset
- OUTPUT FIELDS – Total points, Highscore, headline “LINGO”, headline “Game ended”.
- INPUT FIELD – word entry field

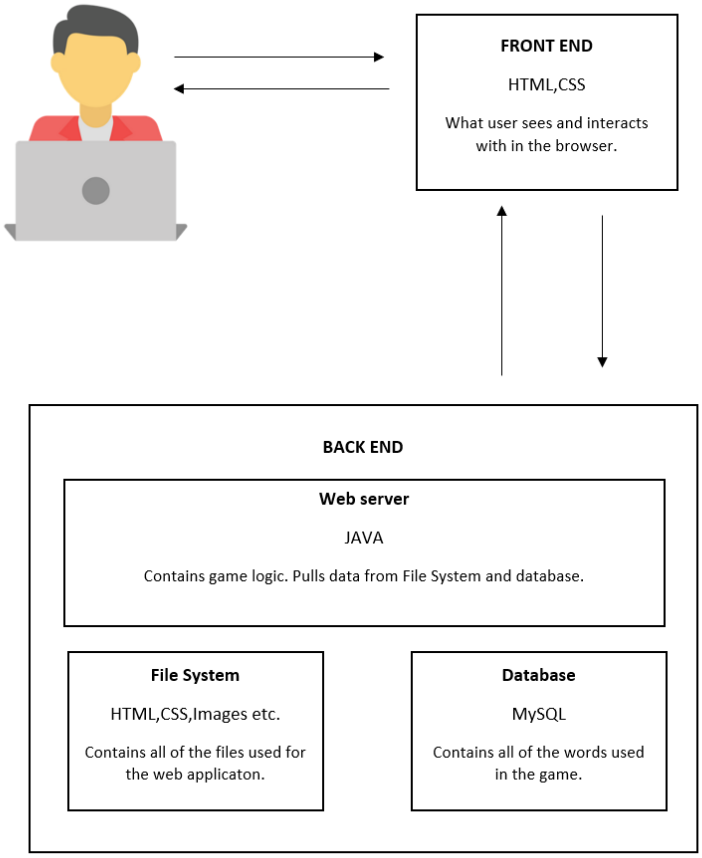
4.2.3. Front-end languages – html, css.

4.2.4. Mobile support is not implemented in this application.

4.2.5. Supported browsers – application needs to support at least these browsers:

- Internet Explorer,
- Google Chrome,
- Mozilla Firefox.

4.3. BACK END design for application:



4.3.1. Back-end languages – JAVA, SQL.

4.3.2. SQL database stores all of the words used in the game. The Server takes words from the database and sends to the game so user can see it.