**Development Software DOS4**

**Create a Serious Game**

**Requirements Document**

**30th May 2018**

**Mr. Makurane**

|  |  |
| --- | --- |
| **Name:** | **Student No:** |
| **Riyaad Cader** | **213117916** |
| **Darren Albertyn** | **214216497** |

Contents

[1. Introduction 2](#_Toc514760312)

[1. Gameplay Overview 3](#_Toc514760313)

[2. Functional requirements 3](#_Toc514760314)

[3. Non-functional requirements 4](#_Toc514760315)

[3.1 Operating Constraints 4](#_Toc514760316)

[3.2 System Constraints 4](#_Toc514760317)

[3.3 Modifiability 4](#_Toc514760318)

[4. Use case model 5](#_Toc514760319)

[5. Class diagram 6](#_Toc514760320)

[6. Dynamic models 7](#_Toc514760321)

[7. User interface – navigational paths and screen mock-ups 8](#_Toc514760322)

[System Design Document 13](#_Toc514760323)

[1. Introduction 13](#_Toc514760324)

[2. System Decomposition 13](#_Toc514760325)

[2.1. Database 13](#_Toc514760326)

[2.2. User Interface 13](#_Toc514760327)

# Introduction

Student Crusher is a game built to test the knowledge of any kid, adult or even the elderly. This game was made to test if users are able to put their brains to the test by selecting a category. The categories consist of Fruits, Continents and Animals. The users will be given random names to guess the complete word.

## Gameplay Overview

The game Students Crush is a game for children of all ages. This is an educational game focused on teaching children the different Continents, Animals and more importantly Fruits. The game allows users to enter their name and requires no login in. Once the user enters their name and selects a category option then clicks the ‘Set Game’ button the user is ready to play the game. A text box is available for users to enter a letter and clicks the ‘Check Letter’ button users are now able to continue guessing until the correct word is displayed. If children are struggling to guess the correct word they are allowed to click the ‘Hint’ button which will allow the user to view a hint and possibly know the word immediately. A tutorial on how to play the game is also listed on the menu.

## Functional requirements

* The user should be able to interact with the system through a designed interface.
* The user should be able to enter their name.
* Based on the category a user selects underscores are representing the letters that should be guessed.
* When user clicks ‘Start Game’ the application requests a random word based in the database.
* If not words can be found in the database an error message will be shown.
* If the user does not enter their name a pop up error will be shown.
* The hint button is available for users offer help.
* The user is able to start a new game at any time.
* The users can click the ‘Check Letter’ to check if the letter exists in the secret word.

Playing the game

* The application should allow a user to enter a letter.
* The letters guessed correctly will be replaced by the input letter.
* If a number is entered or any invalid character an error will pop up.

## Non-functional requirements

## Operating Constraints

* The system will require a WAMP server installed.
* Download the zipped folder which includes ‘Hangman Game’
* Unzip the folder into your www directory.
* From the unzipped folder, import sql database file: ‘GameDatabase.sql’ file using phpMyAdmin.
* Select the Folder containing the Hangman Game which will run automatically
* A web browser is required as well to access the localhost.

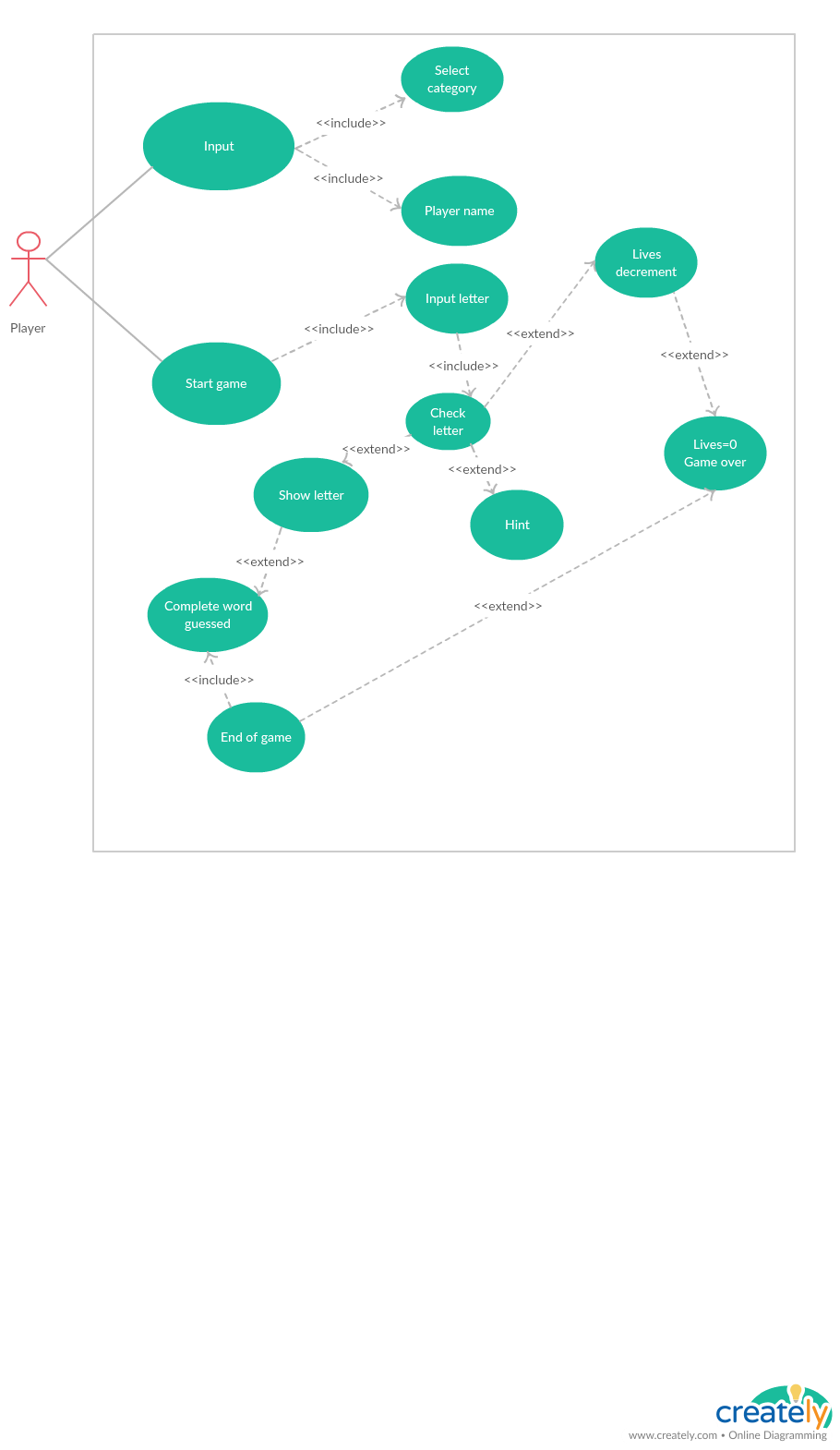
## 3.2 System Constraints

* The program requires a minimum of 2 GB ram and less than 10 megabytes and of hard drive space.
* A standard graphics card is required to run the program.
* The system can run on windows 7, 8 and supports Linux.

## 3.3 Modifiability

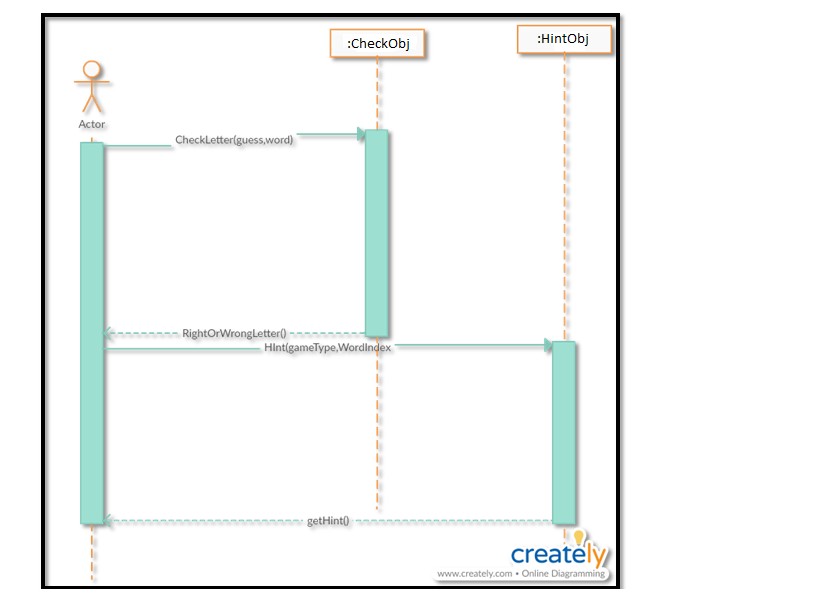
* The word dictionary can be updated by updating the database through phpMyAdmin.
* The user can select different categories of their choice when starting the game.
* The user has a set limit of lives in the game.

## Use case model



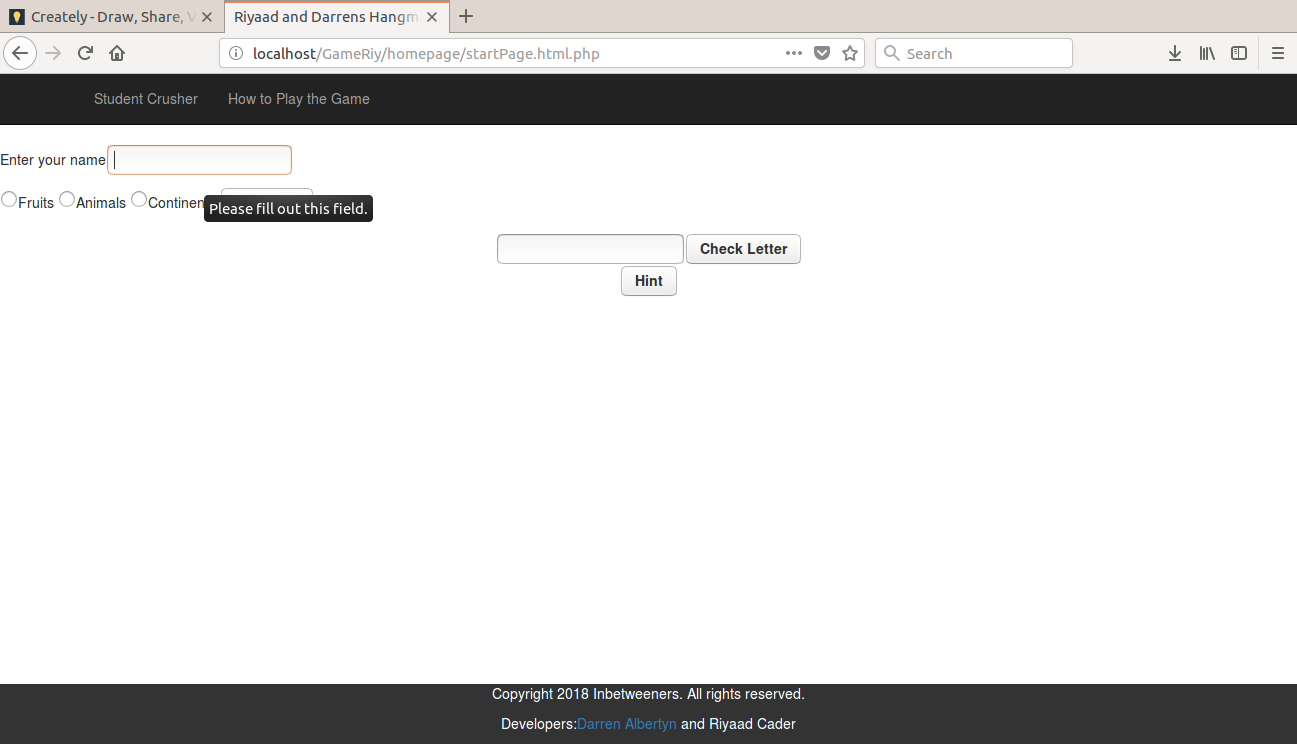
## F:\Speech\StudentCrusherClassDiagram.pngClass diagram

## Dynamic models

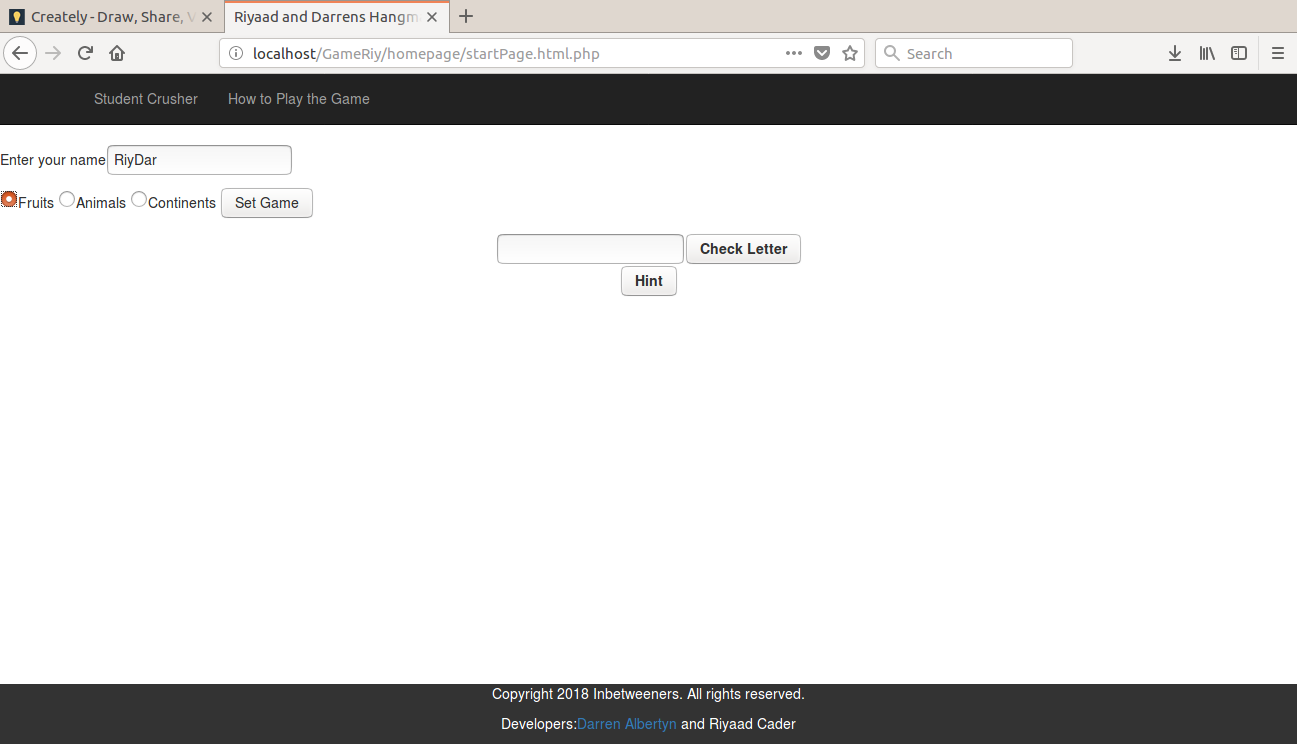


**Figure 1:Scenario :Checking the Letter Function and Clicking the Hint Button**

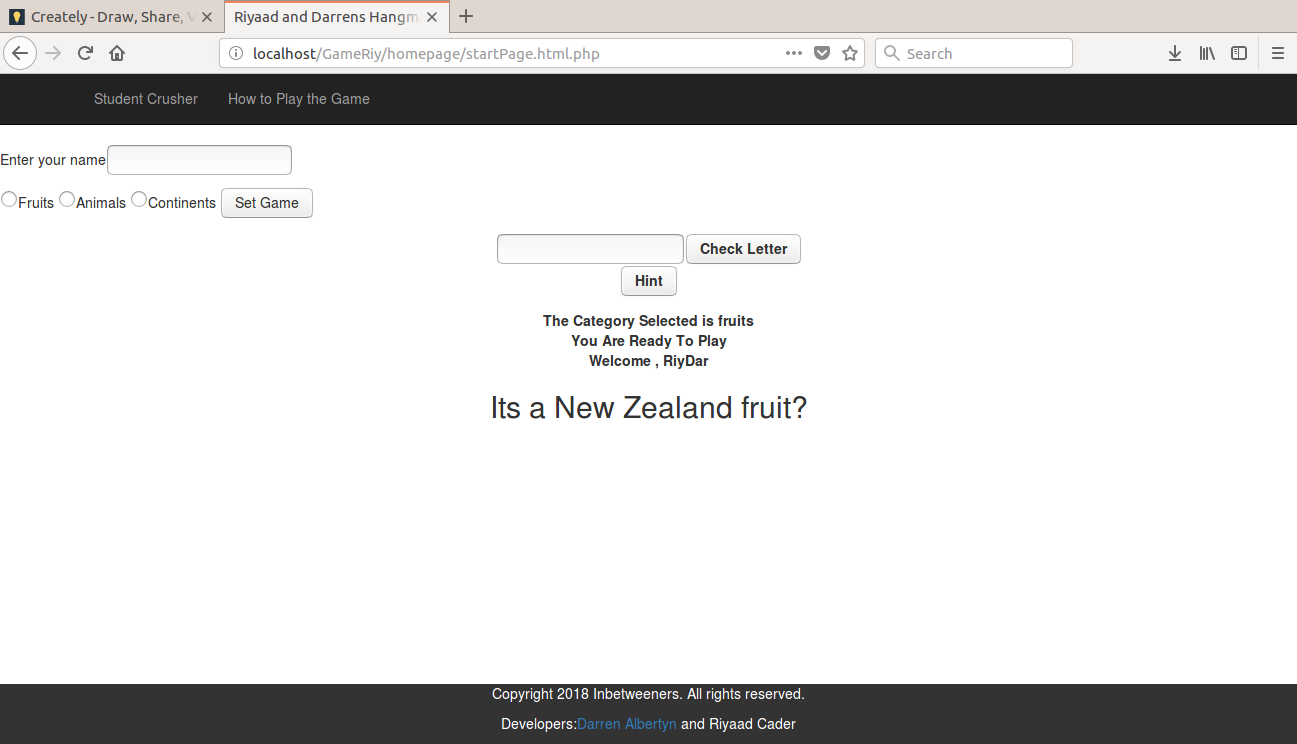
## User interface – navigational paths and screen mock-ups



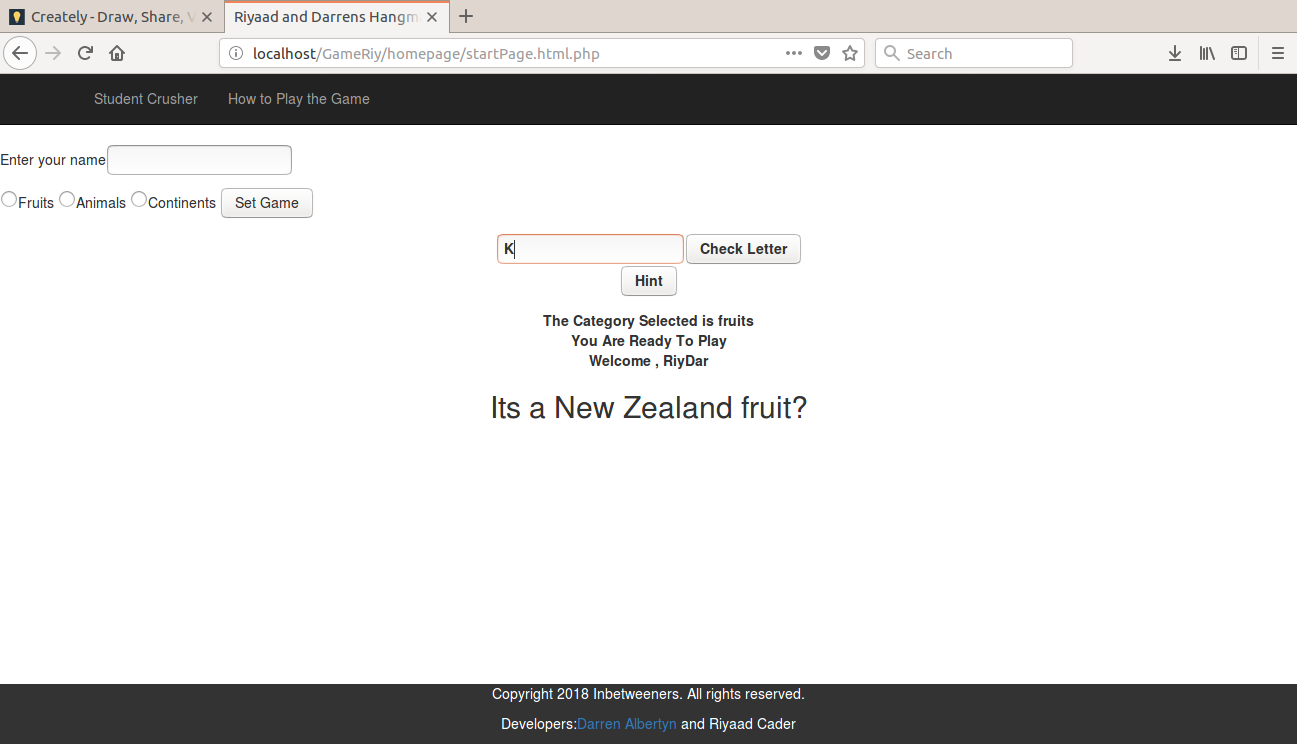
**Figure 2: Home Screen name is required**



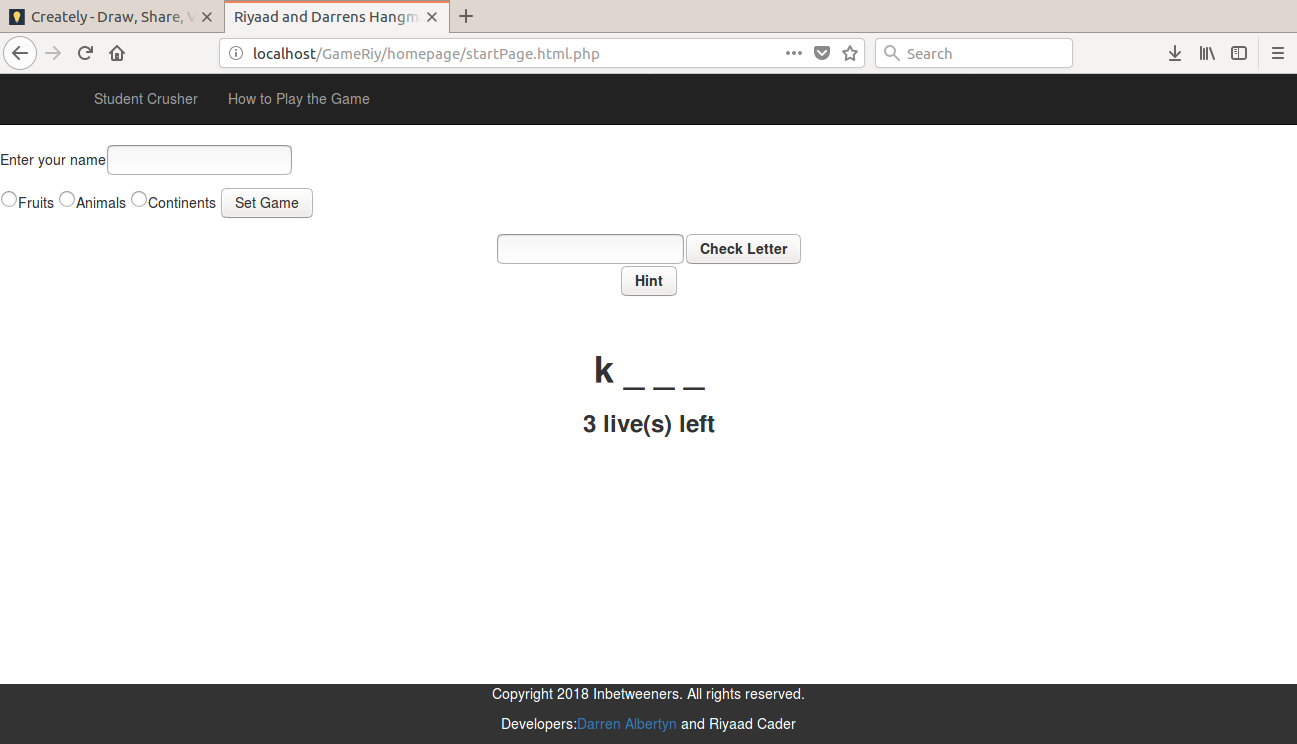
**Figure 3: Name and Category is required and Set Game is clicked**



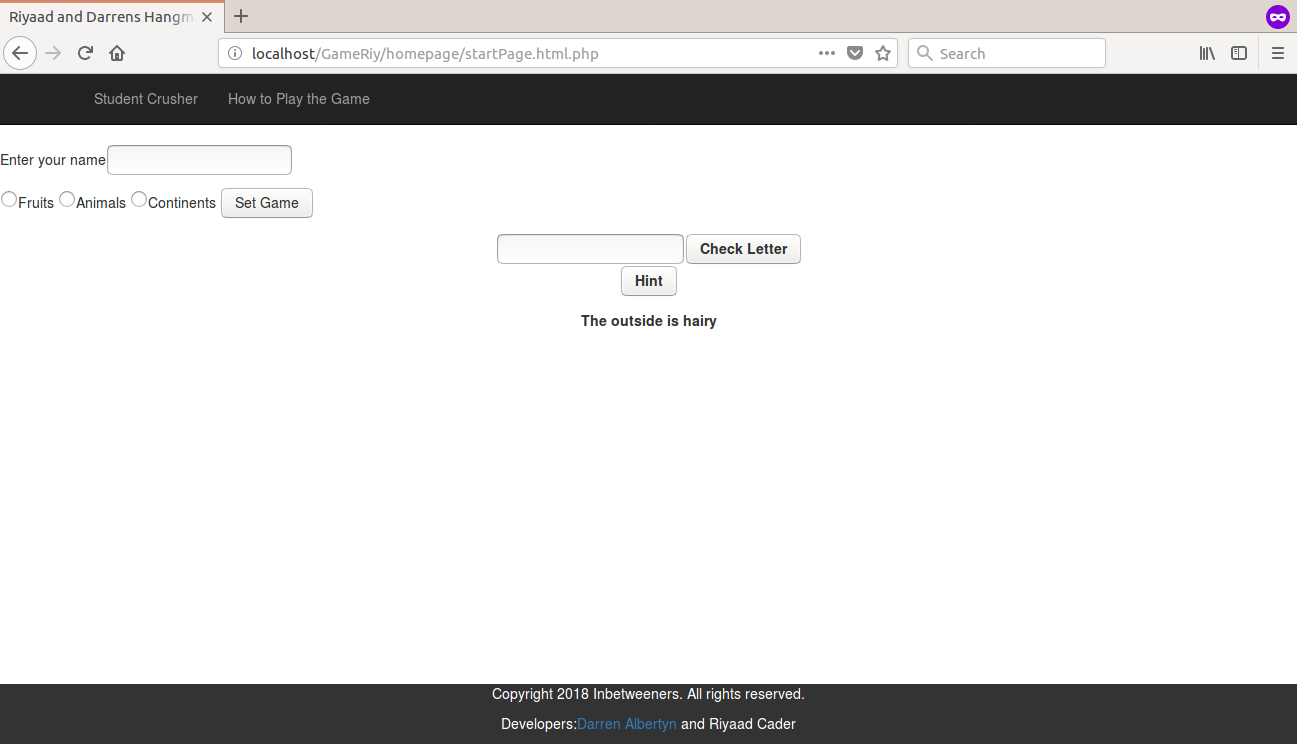
**Figure 4: The Random question is displayed**



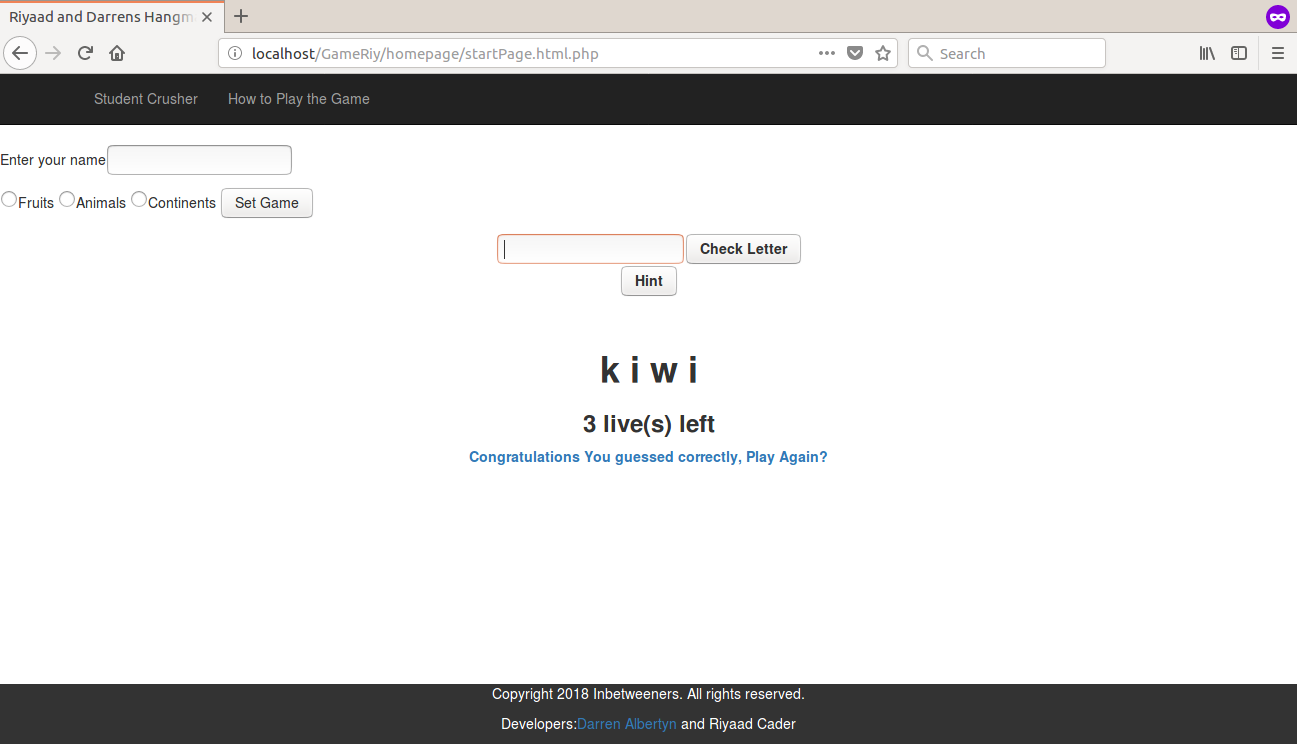
**Figure 5: Guess word by entering Letter**



**Figure 6: The letter is correct**



**Figure 7: The hint Button was clicked**

****

**Figure 8: The word was guessed correctly**

# System Design Document

# Introduction

Student Crusher is a game built to test the knowledge of any kid, adult or even the elderly. This game was made to test if users are able to put their brains to the test by selecting a category. The categories consist of Fruits, Continents and Animals. The users will be given random names to guess the complete word.

## System Decomposition

## Database

The database provides the service for storage purposes. The database retrieves the word dictionary for the game. The database software is phpMyAdmin. The database stores all the questions, hints of words. The tables that exist are the animals, continents and the fruits table. These tables are subdivided into the different categories with the specific data.

## User Interface

The Interface is used to provide communication between the application and the users of the system . The designed is aimed at having different inputs such as the mouse to click different buttons. The keyboard is used to enter different guesses.

