**Documentation for Gaming Website Designing**

**Deepak Kumar Jain**

**CIS 5690 - Advanced Systems Project (Summer, 2019)**

**Dr. Qingxiong Ma**

**January 08, 2020**

**Contents**

[List of Functional Requirements 2](#_Toc29299175)

[Object Domain Model 4](#_Toc29299176)

[Step 1. Identifying Primary List of Objects 4](#_Toc29299177)

[Step 2. Eliminating Duplicated and Unnecessary Items 4](#_Toc29299178)

[Step 3. Building Aggregation Relationships 5](#_Toc29299179)

[Aggregation Relationships 5](#_Toc29299180)

[Step 4. Identify Further Domain Objects 6](#_Toc29299181)

[Step 5. Building Generalization Relationships 6](#_Toc29299182)

[Step 6. Building Domain Model 7](#_Toc29299183)

[Glossary of The Domain Objects: 7](#_Toc29299184)

[Identifying Actors’ List 8](#_Toc29299185)

[Identifying the List of Goals: 8](#_Toc29299186)

[Making List of Use Cases: 8](#_Toc29299187)

[Use Case Diagrams 11](#_Toc29299188)

[Use Case Diagram: User 11](#_Toc29299189)

[Things to keep in mind for use case diagram 13](#_Toc29299190)

[Robustness Diagrams 16](#_Toc29299191)

# **Problem Statement**

It is important to be happy. According to *lifecoach Expert*, - “We become more creative, witty, energetic and fun to be around and it can also lead us to become more financially successful.”

The people can achieve happiness by utilizing their spare time in doing something they like to do; like playing games or gardening or cooking. There are a lot of people in today’s world that feel happy in playing games.

# **Project Scope**

The project should be able to deliver some games that can be played using mouse clicks and keyboard buttons. There should not be any additional hardware required for the user to play the games. There should be a homepage where all the games are hosted and linked to homepage. At the homepage there should be a screen that shows the look and feel of each of the game.

The user should be able to navigate to the individual game pages. When the user clicks on the screens for individual game page, he should be able to navigate to the game page and there must also be links available on the individual game page for the user to navigate among the games and the all pages should be linked to the game homepage.

# **List of Functional Requirements**

Below are the functional requirements of the project.

|  |  |
| --- | --- |
| **Sr. No.** | **Functional requirements** |
| 1 | System will give open access to the website. |
| 2 | User navigates to all the tabs. |
| 3 | **Home** Tab will be welcome Screen to the application. |
| 4 | User can play **FISH** by tapping on the screen. |
| 5 | User can get Score after playing the **FISH** game. |
| 6 | User can play **STRIKE** by playing clicking on the tab. |
| 7 | User can play **STRIKE** by moving the scrollbar at bottom of the game |
| 8 | User will get the **YOU WIN, CONGRATS** Message after game completed |
| 9 | User can play **LUCKY** by playing clicking on the tab. |
| 10 | User need to **Enter your name here to** find his/her luck |
| 11 | User Will get a **Luck** **message** based on this Number of words count |

# **Object Domain Model**

To create the object domain model, we follow the below steps:

**Step 1. Identifying Primary List of Objects**- The below objects are primarily looking useful for the project. First, we will list down all the objects and then in the next step we will eliminate the objects that are not necessary for the project.

**Primary list of objects is as under:**

User

Fish

Strike

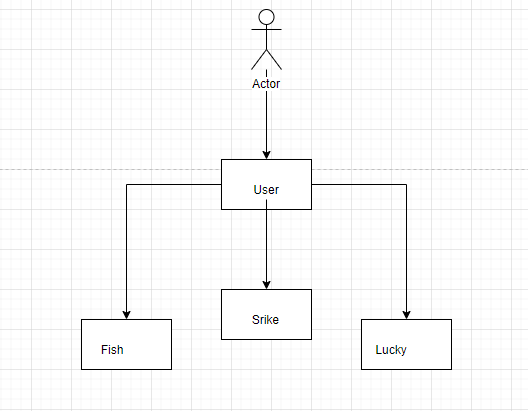
Lucky

**Step 2. Eliminating Duplicated and Unnecessary Items**- In this step we are eliminating duplicate and unnecessary items and making an updated list-

|  |  |
| --- | --- |
| **Identified** | **Eliminated** |
| User  Fish  Strike  Lucky | User Are allowed to use the website. |
| User also play in this website. |

**Step 3. Building Aggregation Relationships-**In this step we are building aggregation relationships in the domain model-

**Aggregation Relationships**



**Step 4. Identify Further Domain Objects**– In this step we identify further domain objects that were not identified in the initial requirements-

Average of individual

Average of team

Average of class

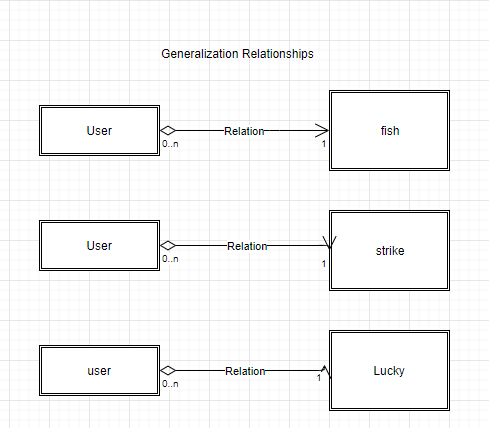
Submit

Retrieve

Plagiarism check

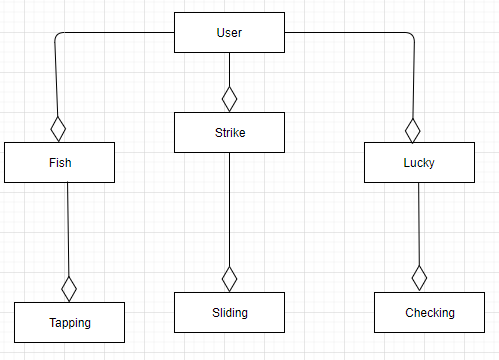
Chat area

**Step 5. Building Generalization Relationships–** In this step we are building generalization relationships in the domain model-



**Step 6. Building Domain Model–** In this step we are building domain model based on both aggregation and generalization relationships-

**Domain Model**



## **Glossary of The Domain Objects:**

**1. User**: User is having three modules to play

**2. Fish**: It’s a game to play by tapping we are going to operate

**3. Strike:** It’s a game to play by Sliding we are going to operate

**4. Lucky;** It’s a game to play by typing the name we are going to check the luck.

**List of Use Cases**

**Identifying Actors’ List:** There is main 1 actor for this project.

1. User

## **Identifying the List of Goals:**

1. **User**

* Login Page for User
* Fish
* Strike
* Lucky
* Contact Us

## **Making List of Use Cases:**

1. User logins to the system
2. Click the fish it will open
3. Click the Strike it will open
4. Click the Lucky it will open
5. Click the Contact Us it will open
6. Use Case Descriptions and Screen Mock-Ups

Full use case description with numbered event flows and screen mock-ups are below:

**Use case 1 -** User logins to the system**:**

**Main flow:**

1. **Primary actor**: User.
2. **Precondition**: User is on the Home page.
3. **Main success scenario**:

A. Click the fish it will open

B. Click the Strike it will open

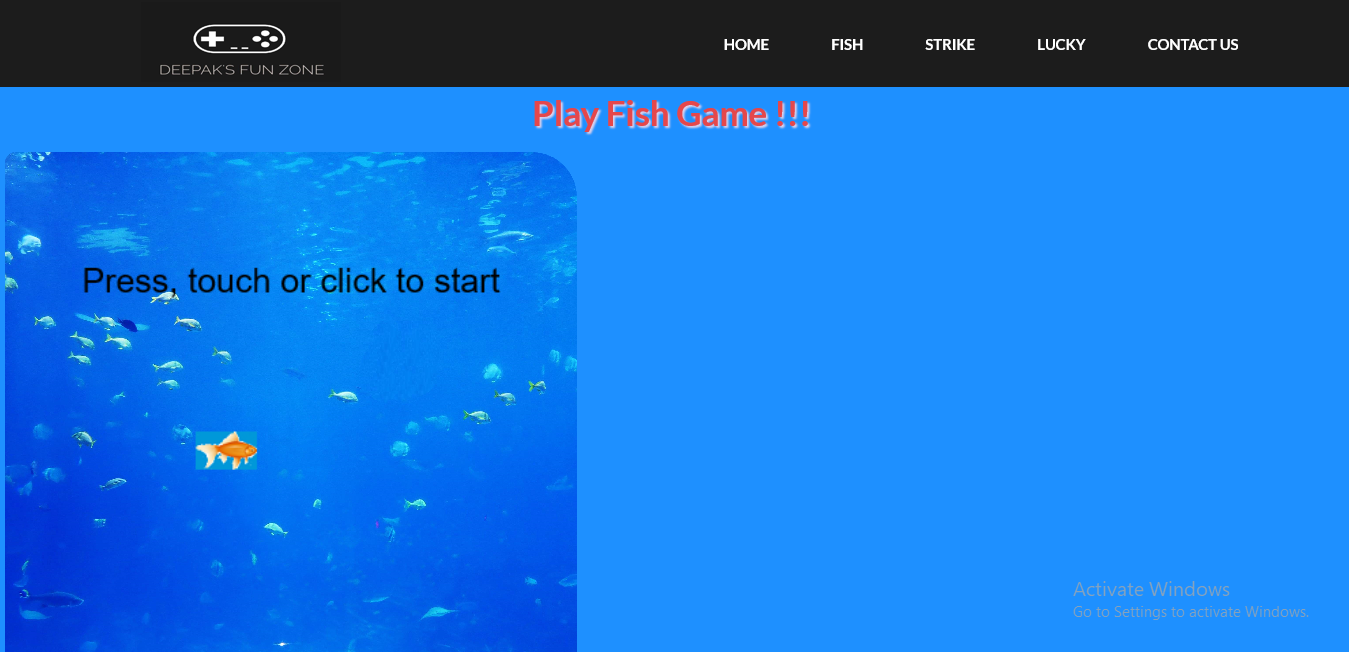
C. Click the Lucky it will open

The user clicks on login link and then the below screens appears. The user provides username, password and selects role and clicks login button.

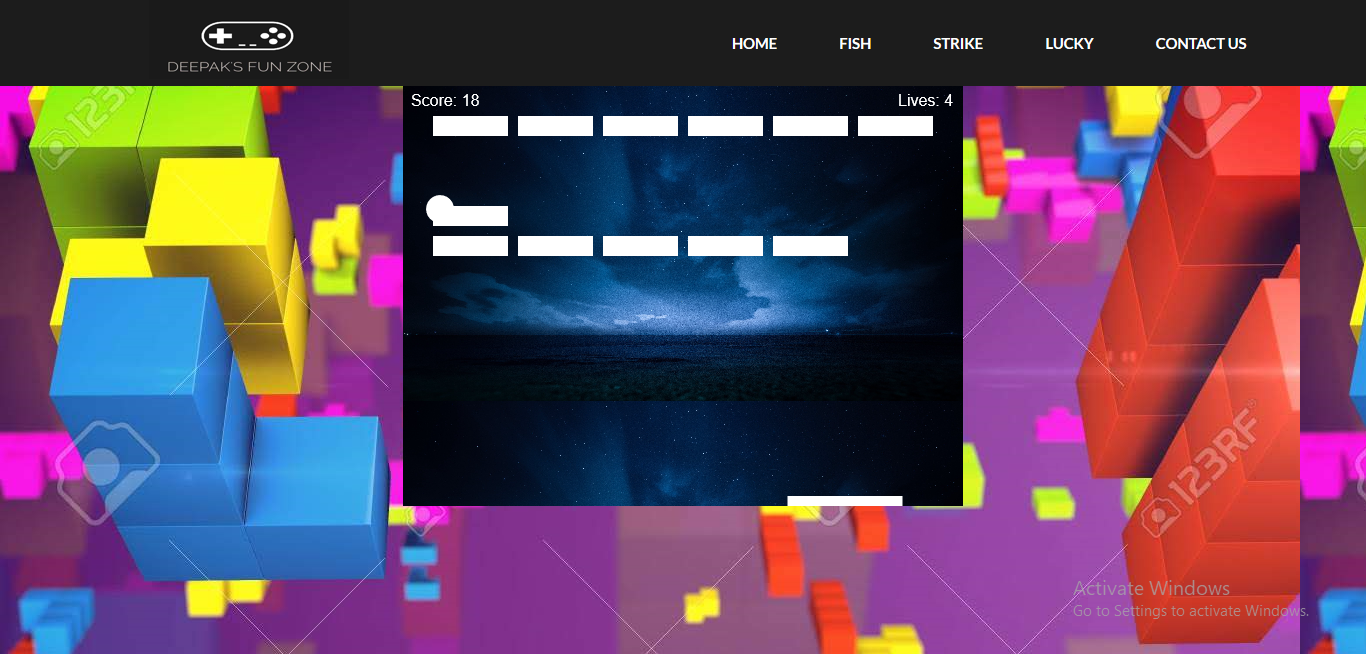
Once the user clicks login button, the system will check the database and shows the below screen if correct username and password are entered.



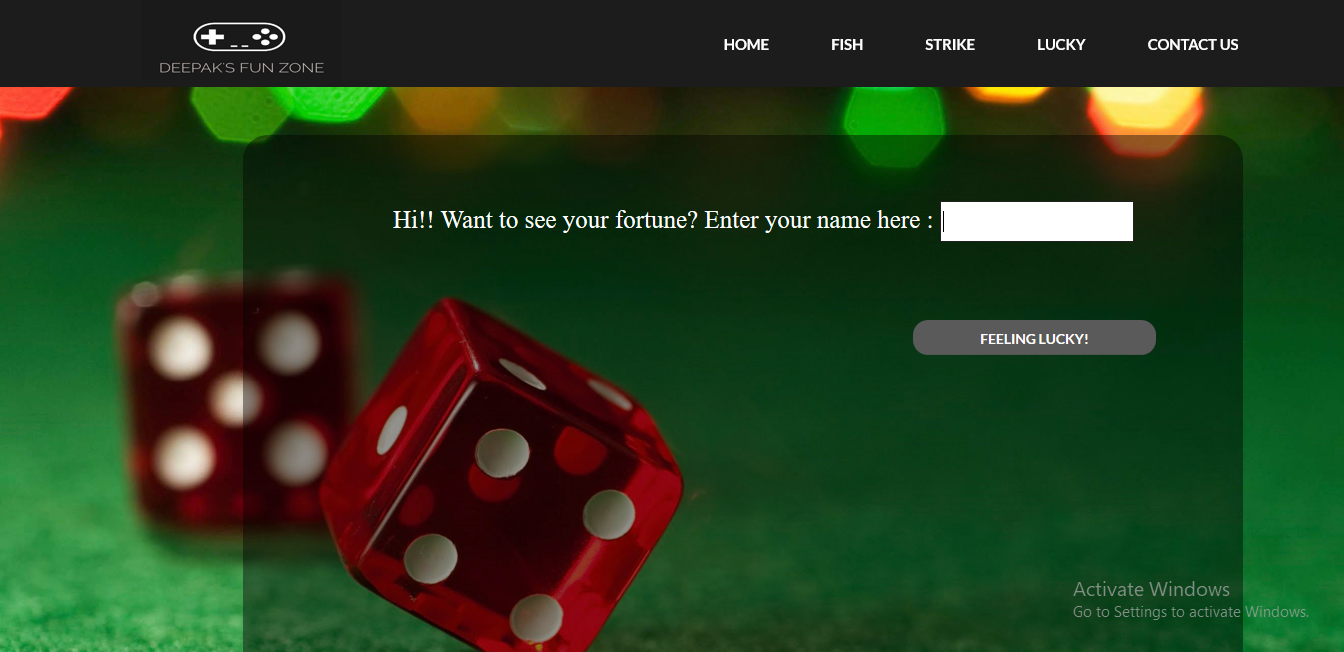
The Home Page of the User it’s a gaming web application



Click the fish it will open, and the game is started when user clicks the mouse button or even presses any of the keyboard buttons.



Click on the Strike game is opens and the game is ready to play.



Click on the lucky the game will open.

You have to give some name, by that name it takes the letters and do the calculation

# **Use Case Diagrams**

## **Use Case Diagram: User**









**Things to keep in mind for use case diagram**

* After the functional descriptions, prototypes and use-cases have been refined, the use case description also must be changes accordingly.
* Check that the use cases should match the prototypes that have been designed.
* All the changes must be incorporated in the use case diagram.

**Activity Diagrams**

Starting below are the Activity diagrams for each use case

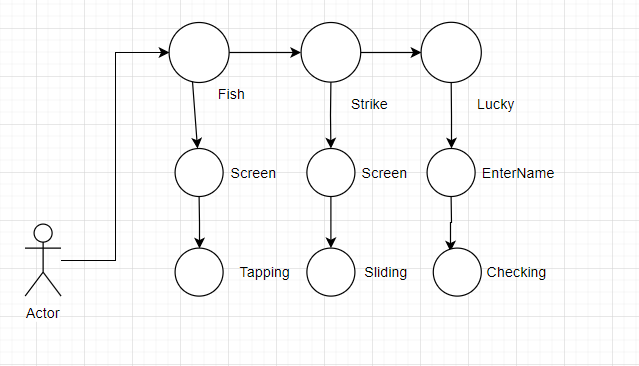






# **Robustness Diagrams**

Starting below are the Robustness diagrams for each use case

****