**Software Requirements and Design Document**

**For**

**Group <13>**

Version 1.0

**Authors**:

Taj Ali

Harege Abay

Tobechukwu Osakwe

Vinson Thomas

# Overview

We are developing a web platform for fellow game gurus to come together and talk about their passions in a homely environment. Our project is an online platform where anyone can write a review of a video game. After reviews have been written, users can comment on reviews, gain reputation/karma points, or search for other reviews. Users can also manage their public profile, setting a profile picture, biographical information, and a link to a personal website or social media. People can vote at most once on a given review.

Users can even create and join communities centered around special topics of their choice. Though the reviews will come out later than proprietary counterparts, they are more likely to be genuine and thorough.

# Functional Requirements

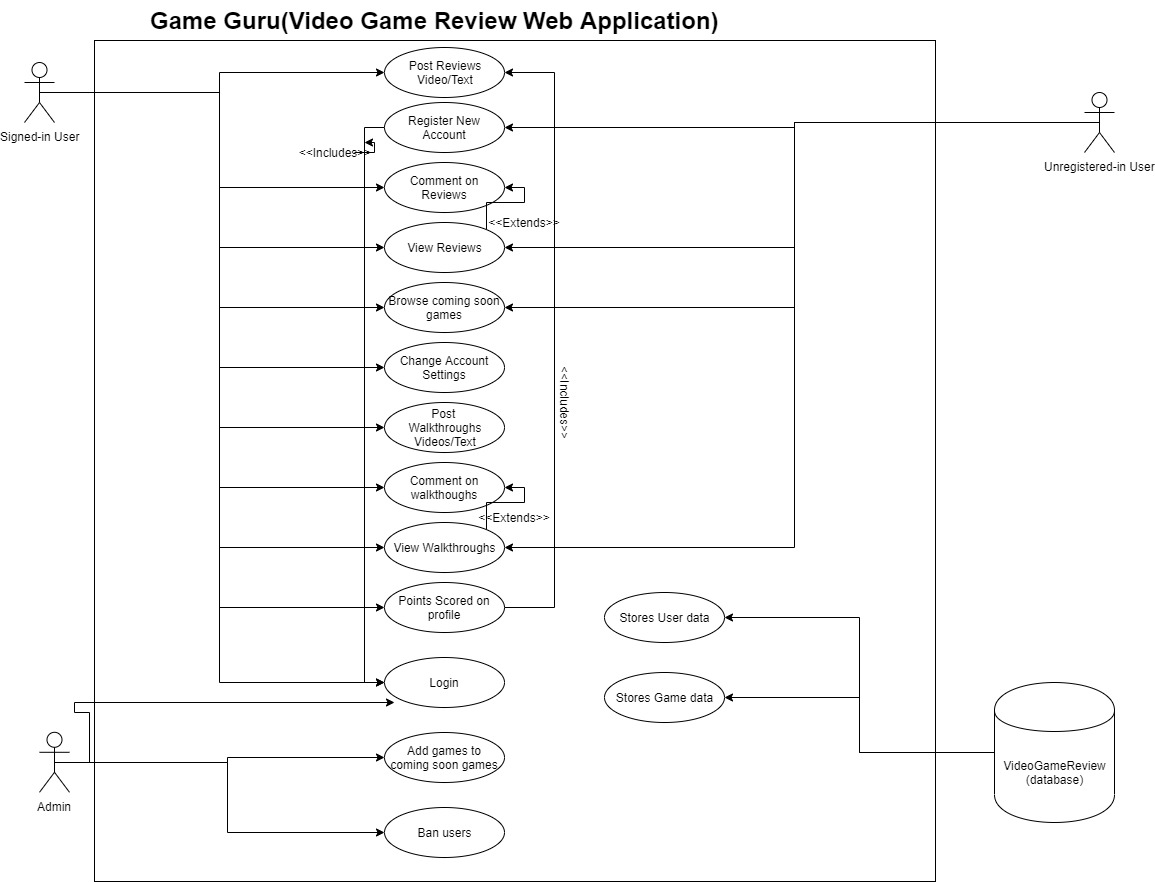
1. *The base website will run and be seen on a web browser.* *(High Priority)*
2. *The system will be able to register user to the website. (High Priority)*
3. *The system will be able to add a user’s bio and profile picture. (medium Priority)*
4. *The system will be able to give the user the ability to review a video game by entering text and giving the game a rating. (High Priority)*
5. *The system will allow the user to post video reviews. (low Priority)*
6. *The system will allow the user to comment on other user’s reviews. (medium priority)*
7. *The system will allow the user to post video game playthrough (Guides to play through certain areas of a game) that will include text. (High Priority)*
8. *The system will allow for the user to post video playthroughs. (low Priority)*
9. *The system will have a web page displaying the games that are coming soon, which will include a release date and a brief description of what that game entails. (High Priority)*

# Non-functional Requirements

*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

1. *In terms of security, Django Hashes passwords allowing for a more secure environment.*

# Use Case Diagram

**

# Class Diagram and/or Sequence Diagrams

*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

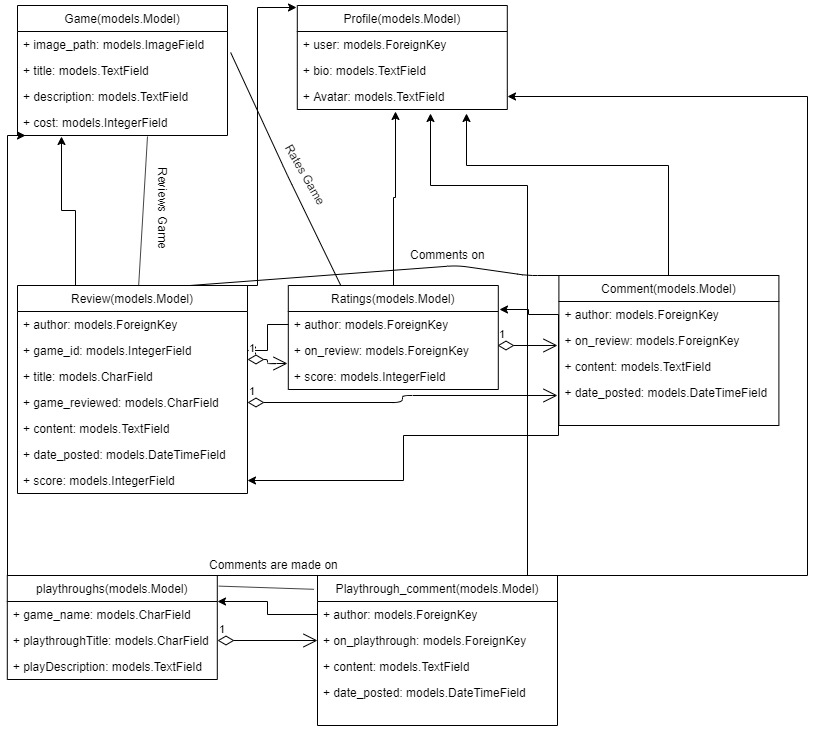
*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

*Class Diagram:*

**

*All classes are used for models in the database.*

# Operating Environment

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

*Software will be pycharm Proffessional and will run in a python environment while also using Django for the web application.*

*Vinson Thomas*

* *Hardware: Windows Laptop running windows 10 Home edition*

*Taj Ali*

* *Hardware: Windows Laptop running windows 10*

*Tobechukwu Osakwe*

* *Hardware: Mac OS Laptop*

*Harege Abay*

* *Hardware: Ubuntu Laptop*

# Assumptions and Dependencies

*We as a group assume that the user has internet and a web browser.*

*We depend on python, Django, Pillow, J-query, and sqlite.*