Contents

[1.0 Introduction 3](#_Toc499316563)

[1.1 Purpose 3](#_Toc499316564)

[1.2 Project Scope 3](#_Toc499316565)

[1.3 Definitions and Acronyms and Abbreviations 3](#_Toc499316566)

[2 User Requirements Definition 3](#_Toc499316567)

[3 Requirements Specification 3](#_Toc499316568)

[3.1 Functional Requirements 3](#_Toc499316569)

[3.1.1 Use Case Diagram 4](#_Toc499316570)

[3.1.2 Requirement 1 – Create user profile 4](#_Toc499316571)

[3.1.3 Requirement 2 - User Login 5](#_Toc499316572)

[3.1.4 Requirement 3 – Add Games 7](#_Toc499316573)

[3.1.5 Requirement 4 – Check Trends 8](#_Toc499316574)

[3.2 Non-Functional Requirements 9](#_Toc499316575)

[3.2.1 Performance/Response time requirements 9](#_Toc499316576)

[3.2.2 Availability requirements 9](#_Toc499316577)

[3.2.3 Recover requirements 9](#_Toc499316578)

[3.2.4 Robustness requirements 10](#_Toc499316579)

[3.2.5 Security requirements 10](#_Toc499316580)

[3.2.6 Reliability requirements 10](#_Toc499316581)

[3.2.7 Maintainability requirements 10](#_Toc499316582)

[3.2.8 Portability requirements 10](#_Toc499316583)

[3.2.9 Extendibility requirements 10](#_Toc499316584)

[3.2.10 Reusability requirements 10](#_Toc499316585)

[3.2.11 Resource requirements 10](#_Toc499316586)

[4 Interface requirements 10](#_Toc499316587)

[4.1 GUI 10](#_Toc499316588)

[4.2 Application Programming Interface 10](#_Toc499316589)

[5 System Architecture 11](#_Toc499316590)

[6 System Evolution 11](#_Toc499316591)

Old Games Corner

# Introduction

# Purpose

The purpose of this document is to outline the requirements of the development of my web service – Old Games Corner.

# Project Scope

The scope of my project is to develop a web service themed towards the collection of old/classic games. The system shall allow users to log in, add games to their collection and also add comments about why they love these games, these ‘memories’ will be displayed in the body of the main page for all users to see. Users may also check the trend of games titles online searches.

# **1.3 Definitions and Acronyms and Abbreviations**

HTML – Hyper Text Mark-up Language

# 2 User Requirements Definition

The system needs to be easily usable with a low clutter UI. Each view of the service will be connected. The system will allow for profile creation and login to the service. The system will allow users to add games to their virtual collection. The system will allow the user to query the trends of retro games on the internet.

# 3 Requirements Specification

A user should be completely comfortable in using all the features of the service after two hours using the system.

# 3.1 Functional Requirements

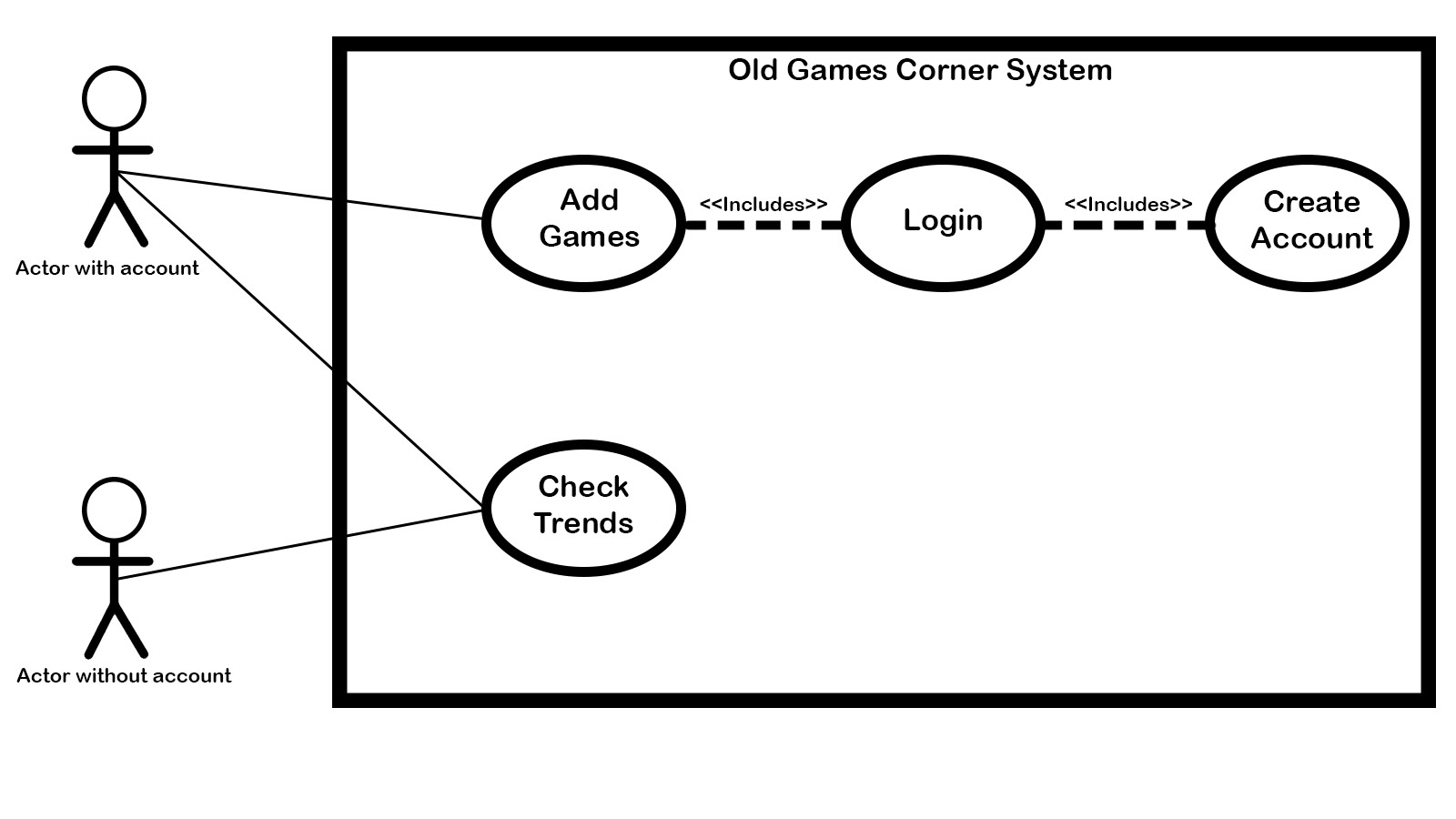
1: The system must allow a user to create an account.

2: The system must allow users to log in to the site using a username and a private password.

3:. The system will allow users to add games to their collection, this comprises of the game’s title, console it was released on, rating, description and also the user’s personal views/reasons for fondness towards the game.

4: The system will allow users to search how frequent a game has been searched online. This feature will take text input from the user and accept it as the game title; the user is required to select a release console for the game via tick-box.

# 3.1.1 Use Case Diagram



# 3.1.2 Requirement 1 – Create user profile

3.1.2.1 Description and Priority

In order for some of the main features of the service to work an actor needs to have profile.

3.1.2.2 Use Case

The actor wants to create a profile on the service in order to access the features.

Scope

Create a user profile on the system that includes a profile name and favourite console. Having a user profile allows actors to add games to their list.

Description

The use case describes how an actor creates a profile with the web service.

Use Case Diagram

Create Account

Flow Description

Precondition:

The System is running and the actor has a device connected to the internet.

Activation

The actor must click the sign up button.

Main flow

1. The user clicks the sign up button.

2. The System displays a window with text fields for the actor to fill out. Each text field has a label; *User Name*, *E-mail*, *Favourite console*, and *Password*. Each field needs to be filled before the actor can proceed. A *Proceed* and *Cancel* button are visible also.

3. The Actor fills out these fields with information and clicks proceed.

4. The System creates a new account.

Alternate Flow

A1 No Alternate Flow.

Exceptional Flow

E1: The Actor clicks proceed without all the fields filled out.

1. The actor clicks proceed but all the fields are not filled out.
2. The system scrolls to the top pf the screen and prompts the Actor to notify them that all the fields must be filled out.
3. The system returns to position 2 of the main flow.

Termination

The Actor clicks the cancel button and the system reverts to the main view of the service.

Post Condition

The system will display the user account page, this displays and extra feature – Add to games collection.

# 3.1.3 Requirement 2 - User Login

3.1.3.1 Description and Priority

This use case is also associated with the functionality of the service. This allows an actor to access their user profile.

3.1.3.2 Use Case

The actor wishes access their account on the service to use the features.

Scope

Actors may login to their user profile so you may add games to your virtual collection.

Description

The use case describes how an actor logs onto a user profile with the web service.

Use Case Diagram

User Login

Flow Description

Precondition:

The System is running and the actor has a device connected to the internet. The Actor must have already created a profile with the site

Activation

The actor must click the login button.

Main flow

1. The user clicks the sign up button.

2. The System displays a window with text fields for the actor to fill out. Each text field has a label; *User E-mail* and *Password*. Each field needs to be filled before the actor can proceed. A *Proceed,* *Cancel* and R*eset password* button are visible also.

3. The Actor fills out these fields with information and clicks proceed.

4. The System authenticates the submission and the profile page is displayed.

Alternate Flow

A1 The Actor has forgotten their password..

1 The Actor click’s on the reset password button.

2 The system then sends an email to their email account. This is a link to reset their password. This page displays two fields, *New Password* and *Confirm new password.*

3 The actor types in their new password and then confirms it in the fields provided.

4 The System confirms the submission and redirects the user to their profile page.

Exceptional Flow

E1: The Actor clicks proceed and both fields do now match.

1. The actor clicks proceed but the fields entries do not match.
2. The System notifies them that the new password does not match with the confirmation new password and redirects them back stage 2 of the main flow.

Termination

The Actor clicks the cancel button and the system reverts to the main view of the service.

Post Condition

The system will display the user account page, this displays and extra feature – Add to games collection.

# 3.1.4 Requirement 3 – Add Games

3.1.4.1 Description and Priority

This allows Actors to add games to their virtual collection and keep track of the games.

3.1.4.2 Use Case

The actor wants to keep a digital listing of their games..

Scope

The scope of this use case is to allow users to add record of their games collection using the service.

Description

The use case describes how an actor adds a game to their collection using the system.

Use Case Diagram

Add Game

Flow Description

Precondition:

The System is running and the actor has a device connected to the internet. The Actor must have already created a profile with the site.

Activation

The actor must click the add game button.

Main flow

1. The user clicks the add game button.

2. The System displays a window with text fields for the actor to fill out. Each text field has a label; *Game Title*, *Console/s, Rating, Description* and *Fond Memories.* Each field needs to be filled before the actor can proceed. A *Proceed*  and *Cancel* button are visible also.

3. The Actor fills out these fields with information and clicks proceed.

4. The System creates a new game in their games collection.

Alternate Flow

A1 No Alternate Flow.

Exceptional Flow

E1: The Actor clicks proceed without all the fields filled out.

1. The actor clicks proceed but all the fields are not filled out.

2. The system scrolls to the top pf the screen and prompts the Actor to notify them that all the fields must be filled out.

1. The system returns to position 2 of the main flow.

Termination

The Actor clicks the cancel button and the system reverts to the main view of the service.

Post Condition

The system will display the users game collection.

# 3.1.5 Requirement 4 – Check Trends

3.1.5.1 Description and Priority

This feature allows the user to search google trends with a query and displays the data.

3.1.5.2 Use Case

The actor wants to view google trends to gain an understanding for the popularity of a game they wish to sell/buy.

Scope

The user queries google trends using a service provided by the site.

Description

This use case describes how a user uses the google trends feature of the service.

Use Case Diagram

Check Trends

Flow Description

Precondition:

The System is running and the actor has a device connected to the internet.

Activation

The actor must click the check trends button.

Main flow

1. The user clicks the check trends button.

2. The System displays a window with a text field for the actor to fill out. This text field has a label; *Game Title*. Next to this are 6 tick box’s each with a label, these are; Sega, *nes*, *snes*, *N64*, *PS1*, and *PS2.* A tick box must be ticked before the actor can proceed. A *Proceed* button is visible also.

3. The Actor fills out the field with information and ticks a box and clicks proceed.

4. The System queries google trends with the info the actor has provided and displays the google trends data on-screen.

Alternate Flow

A1: The user submits a query with only a tick box ticked and no text input to the *Games Title* field.

1. The user submits a query with only a tick box ticked and no text input to the *Games Title* field.
2. The systems submits a query to google trends with only the tick box data as a parameter and the displays the google trends data to the user.

Exceptional Flow

E1: The Actor clicks proceed with only text filled into the *Game Title* field and no tick box selected.

1. The Actor clicks proceed with only text filled into the *Game Title* field and no tick box selected. The system scrolls to the top pf the screen and prompts the Actor to notify them that all the fields must be filled out.
2. The system notifies them that they must tick a box to submit a query. And then to second stage of the main flow.

Termination

The Actor clicks a navigation button that leads to another view of the service.

Post Condition

The system will display the google trends.

# 3.2 Non-Functional Requirements

# 3.2.1 Performance/Response time requirements

The system will be well tested and respond swiftly to user interaction.

# 3.2.2 Availability requirements

Once the system is running all feature are available except the option to add games, this requires user profile and login.

# 3.2.3 Recover requirements

The system will back up its data using GitHub.

# 3.2.4 Robustness requirements

The system will be developed with a high level of standard and will function as required.

# 3.2.5 Security requirements

The system would be protected by web service provider’s security.

# 3.2.6 Reliability requirements

The System will be test thoroughly to ensure it functions correctly.

# 3.2.7 Maintainability requirements

The system will be maintained and checked regularly to ensure it functions as required.

# 3.2.8 Portability requirements

The service will have a responsive front end and be accessible on mobile devices

# 3.2.9 Extendibility requirements

The scope of the system is not extremely vast. Changes to associated app’s will be monitored and changes made accordingly.

# 3.2.10 Reusability requirements

While some systems do have some of the same features as the system in question none combine them into one service.

# 3.2.11 Resource requirements

The majority of the projects resources will be handled within the systems frame work. Other resources such as Google trends data will be acquired.

# 4 Interface requirements

The interface will be clutter free and straight forward. Login and user profile creation are available from the main page. Navigation to the other area of the service are available from the main page too.

# 4.1 GUI

# 4.2 Application Programming Interface

The System will use Google trends API in order for the *CheckTrends* feature.

# 5 System Architecture

The site will developed using python Djangorest framework with html and css on the front end. There will be a model for user, games collection and game for sale, serializers will allow the data to be retrieve from the datasets and pass it to the views.py file in a human readable format. Requests within the views.py file will retrieve the different urls of the service and present them to the user. A plugin called PyTrends will be used for the Google trends feature.

# 6 System Evolution

The system may develop other features along development. These may include a chat section for users, smart trend searches that will store a user’s repeated trend searches and then automatically submit them once the users has logged in.