Requirements Definition

Project4

1. **Introduction and Context**

Project4 is a webpage that aims to allow users to share video games with other users. It will enable users to advertise games that they would like to lend out to others, as well as request video games from other users. Users will be able to search video games based on the number of players the game supports, the console the game is for, and the title, along with other criteria. The general purpose of the application is to allow a person to experience a video game already owned by a friend without the need to purchase another copy of the game. This system will give people an easy to navigate user interface aimed at allowing people to share the video games they love.

The environment for the project will exist primarily as a web application. It will include a user login page, user account page (allowing them to manage games they want to share), a database of available games that users will be able to search for items they might like to borrow from another. This searching feature will have server-side support allowing for users to access the application beyond a localized network; giving users convenient access to their account.

1. **Users and Their Goals**

**Types of users**

* General User (See Figure 2)
  + Goals: Allow searching for games posted by other users
* Owner (See Figure 4)
  + Goals: Allow user to post their games online for sharing
* Admin (See Figure 1)
  + Allow control to delete user posted games if needed
* (System) IGDB API connection
  + Allow system to fetch game details

1. **Functional Requirements**
   1. User Authentication and Access Controls
      1. 1. The system must require all users to authenticate their accounts before giving them access to all features of the system.

1.1.1.  On first login, the user must be able to enter the username and password that a system administrator created for that user.

1.1.2.  Once the user has successful entered the initial username and password, the system must require the user to change the initial password to something of their own choosing.

1.1.3.  For any subsequent login, the user must be able to enter his/her username and current password. If entered correctly, the system must let the user access authorized features of the system. If entered incorrectly, the system should allow the user to try again or exit.

2. User Profile Management

2.1. The system will allow any authenticated user (one who is logged in) to modify his/her own contact information.

2.1.1.  Before being able to change any user contact information the user will have to confirm their identity by re-entering their user name and password.

2.1.2 After User has changed their information a email of conformation will be sent to them through their provided email account

2.2. The system will allow any authenticated user (one who is logged in) to modify his/her own password.

2.2.1. Even while being logged in a change of password will require confirmation from user of prior user name and password.

2.2.2 After User has changed their password a email of conformation will be sent to them through their provided email account

2.3. The system should not allow a user who doesn’t have Admin rights to see or modify any other user profile.

2.3.1 Secure Passwords will be set up so that it is not very easy to access admin privileged accounts.

3. User Add Items Under Logged in Account

3.1. Users will be able to access a GUI to help them upload their various games

3.2. The system should not allow a user who doesn’t have Admin rights to see or modify any other user profile.

3.3. While user uploads a game there will be an API connected to a game database to help with updating correct information

4. Guest/User Browse Inventory

4.1. Anybody with access to the internet should have the ability to browse the database of all games that exist.

4.2 Guests and users will be able to browse the database of information that is sent from the API

5. User Game Request

5.1.Before requesting game, the user will be able to see if it is currently available, or if it is currently checked out.

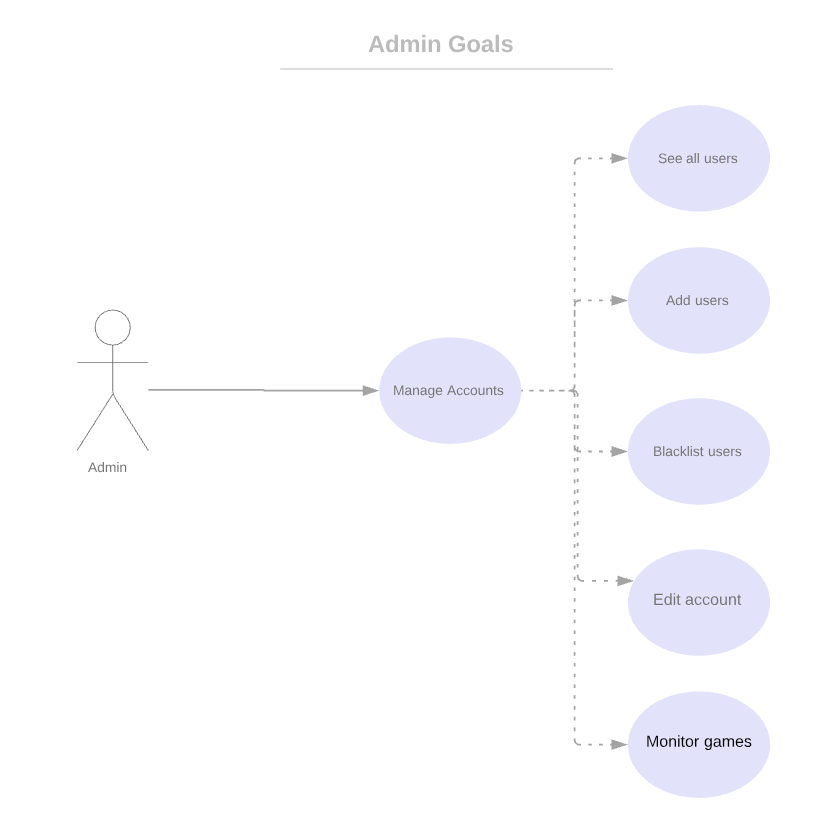
5.1.User will be able to place name on a waiting list that will be stored on the server for safe keepings.

1. **Non-functional Requirements**
   1. The Project4 will be developed using an Agile method with weekly iteration meetings along with consistent communication throughout.
   2. Application-level logic will be tested by each developer including unit tests.
   3. Unit tests should cover at the majority of the codebase.

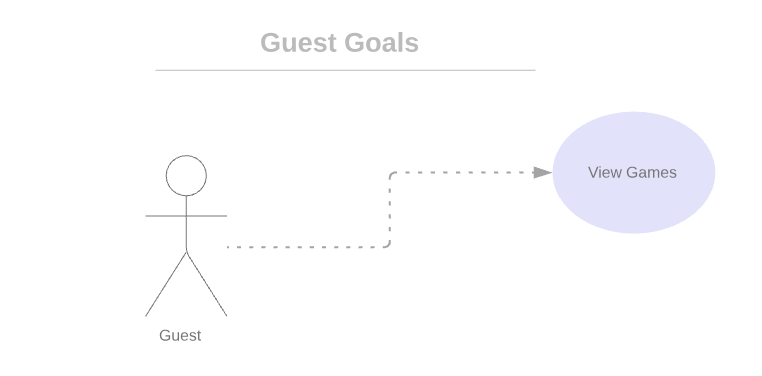
*This section contains requirements that describe and/or constrain the development process. For example, a non-functional requirement may state that development will follow an Agile method with weekly iteration meetings.*

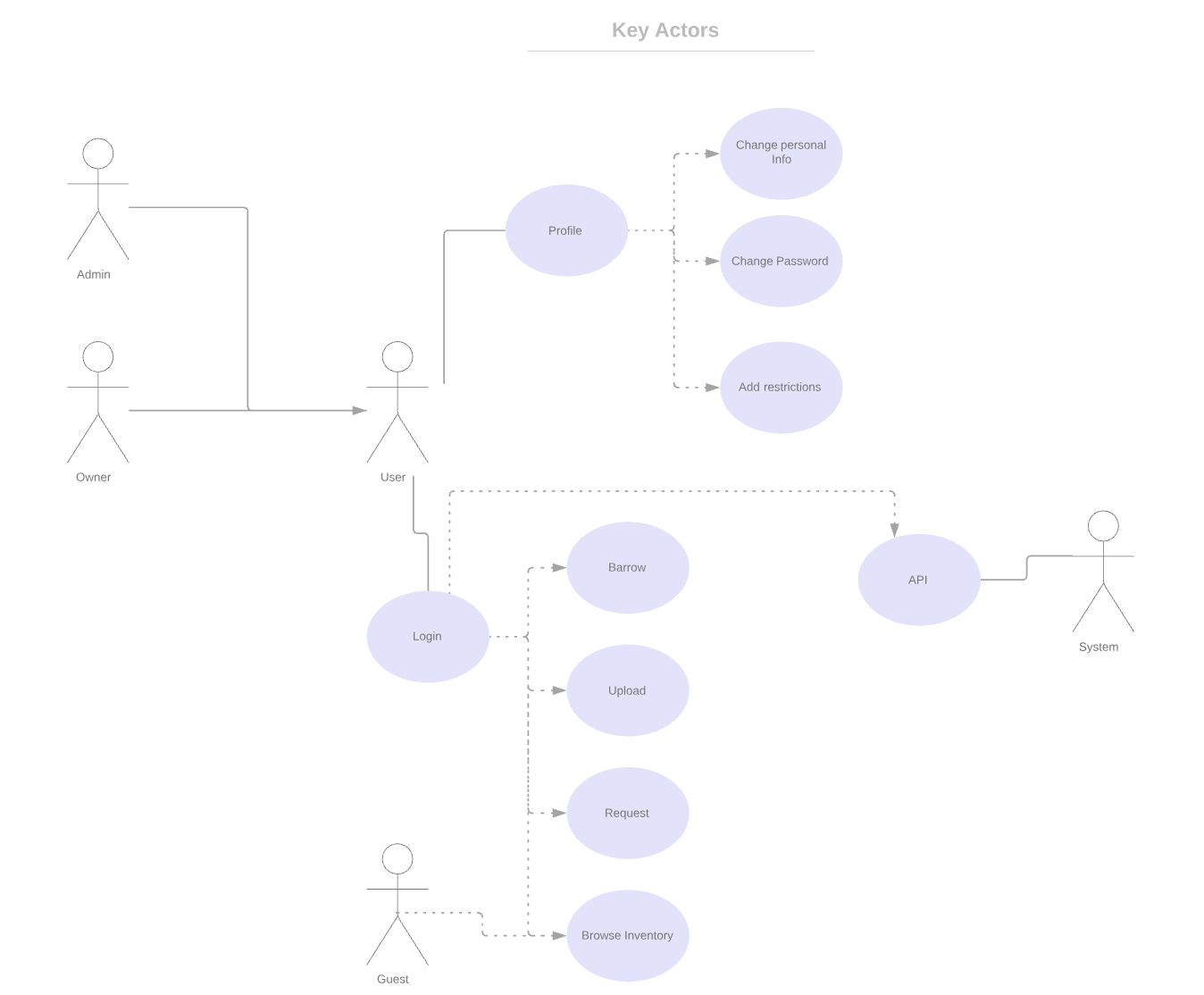
1. **Future Features**
   1. API fetch requests to IGDB to populate game info
   2. Filtering Game searches
   3. Waiting lists for popular games
   4. Visibility Control
   5. User Reviews
2. **Glossary**
   1. *This section contains a list important terms and their definition*
   2. *IGDB: ‘Internet game database’*
   3. *API: ‘application program interface’*
3. **Figures**

**Figure 1.**



**Figure 2.**



**Figure 3.**  

**Figure 4.**

