Principles of Software Engineering, Summer 2023

[FAU: CEN 4010]

Lynx Game Client

Lynx Team (Group 10)

Rodolfo Andres Rivas Matta (Scrum Master) Team Lead - Email: rrivasmatta2021@fau.edu

Mike Clopton (Product Owner) - Email: mclopton2022@fau.edu

Hunter Bresler (Backend Team Lead) - Email: hbresler2022@fau.edu

Cody Smith (Front-End Lead Developer) - Email: codysmith2021@fau.edu

Normil Luccin (GitHub Master) - Email: nluccin2021@fau.edu

Project Site: https://lynxgamestore.com/

GitHub: https://github.com/RARM/Lynx

Youtube Channel: https://www.youtube.com/@LynxGameStore

Youtube Final Demo & Presentation: https://www.youtube.com/watch?v=vdZcG7tadb0

Milestone 5 - Final Project Delivery and Demonstration

August 6, 2023

History Table (Revision Dates)

Revision Date	Details
7/23/2023	**First Draft of Milestone 4** -Added Product Summary -Added Usability Test Plan -Added QA Test Plan -Added Code Review -Added Self-Check on Best Practices for Security -Added Non-Functional Requirements
7/25/2023	**Final Draft of Milestone 4 For Submission** -Completed all necessary sections.
7/28/2023	**First Draft of Milestone 5** **-**Added Product Summary
8/6/2023	**Completion of Milestone 5** **-**Added **Google Analytics Page**

Table of Contents

Product Summary

Lynx is a digital distribution service, game client, and digital storefront looking to put indie game developers into the spotlight and push their game(s) into the masses. This client will be a free software that developers and users can download and used to acquire, download, and launch games directly from their computer. While there exist other digital distribution services used for launching games, such as Steam, Itch.io, and Battlenet, Lynx is unique in that it primarily markets both free and purchasable game titles on its store page. The client includes the following features:

1. **Log in/Sign up Functionality:** Users are able to make an account on the Lynx client from the "Sign Up" page, and then log in to their account to access the client's main pages and functions.
2. **Downloading Games on a "Discover" Store Page:** Users can access the Discover page and add games to their library that can then be accessed on the "Library" tab in the user's client.
3. **Library Page:** Users are able to access their own library that displays all playable games that they have acquired from the Discover page. Users can find games to install on the Discover tab and interact with playable demos from indie developers by finding them in their personal library.
4. **Launching Games:** Users can launch downloaded games from their library and a playable application will be launched from the user's computer.

URL: https://www.lynxgamestore.com

Executive Summary

If you're on the lookout for a top-notch digital distribution service storefront for video games - particularly indie games - then you needn't look any further than Lynx. Built with a priority on independent creators' works by providing an extensive array of titles, primarily targeted towards PC gamers or anyone interested in playing indie or third-party software.

Lynx has four principal sections: The store section features tailored filters for users' personalized recommendations based on their liking, while allowing them to purchase new games at ease. The library section enables efficient organization and tracking owned content, alongside providing ample room for exploration. Community section allows users the ability to participate in forums, discussion groups or offer feedback directly designed by game creators who actively monitor it themselves. Last but not least, the user profile section makes customization and personalization truly accessible.

Generating revenue purely from game sales and optional purchasable cosmetics alone; Lynx refrains from overcrowding its platform by shunning advertisements or subscription services, which can sometimes spoil customer perception of the platform. It will be a native platform on only Windows PC; however, there are plans for potential expansion in due course, allowing other Operating Systems to be used to run Lynx.

What sets Lynx apart is its emphasis on supporting game creators of all sizes and building communities around games and computer software. All donations made to game creators using fundraising features will go directly to them, with Lynx only profiting from cosmetics and a portion of game sales in the store. Lynx’s primary goal will be to attract indie developers and fans of indie games by providing an easy-to-enter market with a wide array of features and promotional tools to efficiently market games to users from within the client.

Lynx, a platform that respects and supports both users and creators, fostering a loyal and happy community.

Use Cases

Use Case - Purchasing Indie Games

The user logs into the Lynx platform and browses through the store. They find an indie game they are interested in by browsing to the indie game section, view its details, and decide to purchase it.

1. **Description:** This use case describes the process of how a user will purchase an indie game on the Lynx platform.
2. **Actors:**
    2.1. User
    2.2. Lynx Platform
3. **Preconditions:**
    3.1. User has an active Lynx account and is logged in via Wi-Fi.
    3.2. The platform is available.
4. **Primary Flow of Events:**
    4.1. User logs into the Lynx platform.
    4.2. User navigates to the indie games section on the store page.
    4.3. User selects an indie game they are interested in.
    4.4. User views the game details.
    4.5. User decides to purchase the game and completes the transaction.
    4.6. Game appears in the user's library and downloaded on computer.
    4.7. Terminate Use Case - Purchasing Indie Games.
5. **Alternate Flows:**
    5.1. User is not logged into Lynx account (from Primary Flow 4.1)
        5.1.1. System asks for user's login information
            5.1.1.1. If user has not created an account, go to Alternative Flow 5.2
        5.1.2. User inputs login information
        5.1.3. System validates login information
            5.1.3.1. If login information is invalid, return to step 5.1.1

5.1.4. End Alternative Flow 5.1 - Return to Primary Flow, step 4.2

5.2. User has not created an account yet (from Alternative Flow 5.1.1.1)

5.2.1. System asks user to make an account

5.2.1.1. If user doesn't wish to create account, End Alternative Flow 5.2, return to primary flow 4.7

5.2.2. User pressed the Create Account button

5.2.3. An account creation form is provided to the user by the system

5.2.4. User fills in form and presses Submit button.

5.2.5. System creates a new member account and enters it into the database

5.2.6. End Alternative Flow - return to Primary Flow, step 2

5.3. User does not want current game (from Alternative Flow 4.4)

5.3.1. User clicks on Back button or returns to store page

5.3.2. End Alternative Flow - return to Primary Flow, step 3

5.4. User payment information is invalid (from Alternative Flow 4.5)

5.4.1. System asks user to re-enter payment information or change payment method

5.4.1.1. If user payment methods are continuously declined and the user cannot complete the transaction, End Alternative Flow 5.4, return to primary flow 4.7

5.4.2. System verifies payment information and the transaction is completed.

5.4.3. End Alternative Flow - return to Primary Flow, step 6

Use Case - Game Developer Features

A game developer logs into the Lynx platform and accesses their game page, utilizing a variety of features ranging from editing and publish game data to exploring community and marketing options.

1. **Description:** This use case describes the process of how a game developer will access features to support the creation and managing of their video game or software.
2. Actors:
    2.1. Game Developer
    2.2. Lynx Platform
3. Preconditions:
    3.1. Game Developer has an active Lynx Developer Account and is logged in via Wi-Fi.
    3.2. The Developer has published or is in the process of publishing a game.
    3.3. The platform is available and functioning.
4. Primary Flow of Events:
    4.1. Game Developer logs into the Lynx Platform via Developer Account.
    4.2. Game Developer navigates to their game's page.
    4.3. Game Developer selects the feature desired to add or edit for game (e.g., technical support, creating community post, publishing game changes, and creating fund-raisers for the game)
    4.4. Game Developer presses Submit button to save changes to the Lynx database.
    4.5. System saves changes to the Lynx database.
    4.6. Terminate Use Case - Game Developer Features
5. Alternative Flows:
    5.1. Developer is not logged into Lynx account (from Primary Flow 4.1)
        5.1.1. System asks for Developer's login information
            5.1.1.1. If user has not created an account, go to Alternative Flow 5.2
        5.1.2. Developer inputs login information
        5.1.3. System validates login information
            5.1.3.1. If login information is invalid, return to step 5.1.1
        5.1.4. End Alternative Flow 5.1 - Return to Primary Flow, step 4.2
    5.2. Developer has not created an account yet (from Alternative Flow 5.1.1.1)
        5.2.1. System asks Developer to make an account

5.2.1.1. If Developer doesn't wish to create account, End Alternative Flow 5.2, return to primary flow 4.6

5.2.2. Developer pressed the Create Account button

5.2.3. An account creation form is provided to the user by the system

5.2.4. Developer fills in form and presses Submit button.

5.2.5. System creates a new account and enters it into the database.

5.2.6. End Alternative Flow - return to Primary Flow, step 2

5.3. Developer has not created a game or software yet (from Alternative Flow 4.2)

5.3.1. Developer clicks on Create Game button

5.3.1.1. If Developer doesn't wish to create a game, End Alternative Flow 5.3, return to primary flow 4.6

5.3.2. System asks Developer to fill in Game Creation form

5.3.3. System asks Developer to upload game files to Lynx database

5.3.4. Lynx staff receives the Developer's game form and files to review and verify

5.3.4.1. If Lynx staff does not approve of the game submission, return to step 5.3.1

5.3.5. Lynx staff approves of the game and uploads it to the database and store page

5.3.6. The Developer's game becomes editable on the game page

5.3.7. End Alternative Flow - return to Primary Flow, step 3

Use Case - Launching and Playing Games from Library

The user logs into the Lynx platform to play a game that they have purchased off the Lynx store and own in their Game Library. They click on one of the games listed in their Library section, access the game details, then press the Play button to launch it on their computer.

1. **Description:** This use case describes the process of how a user will purchase an indie game on the Lynx platform.
2. **Actors:**

2.1. User

2.2. Lynx Platform

3. Preconditions:

3.1. User has an active Lynx account and is logged in via Wi-Fi.

3.2. The platform is available and functioning.

3.3. User owns at least one game purchased from the Lynx Store

4. Primary Flow of Events:

4.1. User logs into the Lynx platform

4.2. User navigates to Game Library page.

4.3. User selects game they want to play

4.4. User clicks on the Play button to launch the game

4.5. Lynx Platform launches the game

4.6. Terminate Use Case - Launching and Playing Games from Library

5. Alternative Flows:

5.1. User is not logged into Lynx account (from Primary Flow 4.1)

5.1.1. System asks for user's login information

5.1.1.1. If user has not created an account, go to Alternative Flow 5.2

5.1.2. User inputs login information

5.1.3. System validates login information

5.1.3.1. If login information is invalid, return to step 5.1.1

5.1.4. End Alternative Flow 5.1 - Return to Primary Flow, step 4.2

5.2. User has not created an account yet (from Alternative Flow 5.1.1.1)

5.2.1. System asks user to make an account

5.2.1.1. If user doesn't wish to create account, End Alternative Flow 5.2, return to primary flow 4.6

5.2.2. User pressed the Create Account button

5.2.3. An account creation form is provided to the user by the system

5.2.4. User fills in form and presses Submit button.

5.2.5. System creates a new member account and enters it into the database

5.2.6. End Alternative Flow - return to Primary Flow, step 4.2

5.3. User has lost access to previously owned game in Library due to refunding (from Alternative Flow 4.3)

5.3.1. System redirects user back to the store page for the game to purchase again in order to play

5.3.1.1. If the user does not wish to purchase the game, return to step 4.2

5.3.2. User decides to Purchase the game, go to Use Case - Purchasing Indie Games Primary Flow step 4.5

5.3.3. End Alternative Flow - return to Primary Flow, step 4.4

Use Case - Accessing Gaming Communities

The user logs into the Lynx platform and navigates to the community section. They browse through the available gaming communities and join one that matches their interests.

1. **Description:** This use case describes the process of how a user will join a gaming community on the Lynx platform.
2. **Actors:**
   - 2.1. User
   - 2.2. Lynx Platform
3. **Preconditions:**
   - 3.1. User has an active Lynx account and is logged in via Wi-Fi.
   - 3.2. The platform is available and functioning.
4. **Primary Flow of Events:**
   - 4.1. User logs into the Lynx platform
   - 4.2. User navigates to the community section
   - 4.3. User browses through the available communities
   - 4.4. User selects a community that aligns with their interests and joins it.
   - 4.5. The database adds the user to the list of community members, allowing the user to participate in discussions and events.
   - 4.6. Terminate Use Case - Accessing Gaming Communities

5. **Alternative Flows:**
    5.1. User is not logged into Lynx account (from Primary Flow 4.1)
        5.1.1. System asks for user's login information
            5.1.1.1. If user has not created an account, go to Alternative Flow 5.2
        5.1.2. User inputs login information
        5.1.3. System validates login information
            5.1.3.1. If login information is invalid, return to step 5.1.1
        5.1.4. End Alternative Flow 5.1 - Return to Primary Flow, step 4.2
    5.2. User has not created an account yet (from Alternative Flow 5.1.1.1)
        5.2.1. System asks user to make an account
            5.2.1.1. If user doesn't wish to create account, End Alternative Flow 5.2, return to primary flow 4.6
        5.2.2. User pressed the Create Account button
        5.2.3. An account creation form is provided to the user by the system
        5.2.4. User fills in form and presses Submit button.
        5.2.5. System creates a new member account and enters it into the database
        5.2.6. End Alternative Flow - return to Primary Flow, step 4.2

Use Case - Editing Profile

The user logs into the Lynx platform and navigates to the profile section. The user is able to view and edit their profile, which can be viewed by friends on the platform.

6. **Description:** This use case describes the process of how a user will be able to edit their profile for others to see.
7. **Actors:**
    7.1. User
    7.2. Lynx Platform
8. **Preconditions:**
    8.1. User has an active Lynx account and is logged in via Wi-Fi.
    8.2. The platform is available and functioning.

9. Primary Flow of Events:

9.1. User logs into the Lynx platform

9.2. User navigates to the profile section

9.3. User is able to view their current profile.

9.4. The user is able to place their favorite games, achievements, and purchased cosmetics onto their profile for others to see.

9.5. The database updates the user's profile when the changes are saved by the user.

9.6. Terminate Use Case - Editing Profile

10. Alternative Flows:

10.1. User is not logged into Lynx account (from Primary Flow 4.1)

10.1.1. System asks for user's login information

10.1.1.1. If user has not created an account, go to Alternative Flow 5.2

10.1.2. User inputs login information

10.1.3. System validates login information

10.1.3.1. If login information is invalid, return to step 5.1.1

10.1.4. End Alternative Flow 5.1 - Return to Primary Flow, step 4.2

10.2. User has not created an account yet (from Alternative Flow 5.1.1.1)

10.2.1. System asks user to make an account

10.2.1.1. If user doesn't wish to create account, End Alternative Flow 5.2, return to primary flow 4.6

10.2.2. User pressed the Create Account button

10.2.3. An account creation form is provided to the user by the system

10.2.4. User fills in form and presses Submit button.

10.2.5. System creates a new member account and enters it into the database

10.2.6. End Alternative Flow - return to Primary Flow, step 4.2

Data Definition

- **Storefront.** It is a digital store. In this context, a store for games and other software available on the platform.
- **Digital Distribution Service.** It is a service (software) for distributing goods or services. In this context, our storefront is a digital distribution service.
- **Lynx Platform.** It is the set of the Lynx server, website, and client. Together, they provide the service of purchasing and accessing indie games.
- **Lynx Server.** This component is the backend module (living in a server) that serves as the central system. It keeps track of all user, developer, game, and community records.
- **Lynx website.** The Lynx website is just a web location that allows users to download the client. It does not provide any other service. Navigating, purchasing, or installing games (including using communities) requires the Lynx client.
- **Lynx client.** It refers to the client application installed in the user and developer's computer. It is a service that allows the user to purchase and download games. They can also use communities and launch their games (purchased from Lynx) there. Developers can manage and upload their games through the Lynx client as well.
- **Indie Games.** They are games created independently by one person or small teams without the financial support of large corporations.
- **AAA Games.** These are games created by mid-size or large game publishers. These have a more extensive budget and teams.
- **Game Developer.** It refers to a small team or a single person (actor) considering publishing games they create on the Lynx platform.
- **User.** This actor is the person who will be using the platform to purchase and download games. They may also use the communities to engage with other actors with the same interests.

- **Lynx Account.** It represents (data) the system's most basic actor (neither a Lynx staff nor a developer). It can have minimal interaction with the system (purchasing, downloading, playing games, and using communities). They could also create and manage communities they own.
- **Developer Account.** It represents an actor with more privileges than the usual Lynx account. A developer account can also upload and manage games. Moreover, they can also manage communities linked to their games in addition to standard communities.
- **Lynx Staff Account.** This account represents the actor with the most privileges in the system. They approve games and serve as moderators on the Lynx platform.
- **Library.** Every Lynx Account (including developers and staff) has a library linked to the account. The library is a collection of games purchased by the actor.
- **Game.** It refers to the computer software sold through the Lynx platform. Developers can publish them, and users can buy them. Games can also have communities linked to them.
- **Community.** It is a service for developers and typical users to engage. It allows them to share messages and experiences.
- **Search.** The Lynx system provides this service through the Lynx client. Actors can search games by name or description.
- **Filter.** It is a service that works with the search or on the user's homepage. It provides a way of filtering a list of games to match specific characteristics (e.g., genre, price, tags) or user preferences.
- **User preferences.** It is the data in the Lynx system about the user linkings based on their input or behavior playing games.

List of Functional Specifications

1. Lynx Account System/Sign up and Login

1.1. User should be able to sign up for an account on Lynx. To sign up, the user would enter their username, email, password, phone number, DOB, full name, gender and security question(s). If a user forgets their username and password, they can try to recover their account using their email, to which their username would be displayed, and they would be required to reset their password. A user can not create an account if any of the fields mentioned above already exists in the system's database. User also can not create an account if any of the required fields is empty. The required fields are First name, Last name, Username, Password, Email, Date of Birth (DOB), Phone Number and/or Security Question and Security Answer.

1.2. Sequence

1.2.1. User enters first name

1.2.2. User enters last name

1.2.3. User enters DOB

1.2.4. User enters username

1.2.5. User enters email

1.2.6. User enters password

1.2.7. User re-enters password for confirmation

1.2.8. User can either enter their phone number and/or enter a security question and security answer

1.2.9. System will check if the Username is available

1.2.10. System will validate Password

1.2.11. System will store their Full name, Phone number and/or answer to selected security question

1.2.12. System will confirm to the user that the account was created

1.2.13. System will redirect them back into the login page

1.3. Function requirement label

1.3.1. REQ 1.1 Login/Account System

2. Library

2.1. Users and Developers that have a Lynx account will have a library of owned games from both Lynx and other DDSs that they can play from.

2.2. Sequence

2.2.1. User/Developer is automatically redirected to the Library page

2.2.2. User/Developer selects a game to play of their choosing

2.2.3. System activates the client and launches the game. If the game is from another DDS, the system launches the DDS first and then launch the game from the DDS

3. Discover

3.1. User should be able to access a page where upcoming and popular indie games are displayed and shown for users to engage in.

3.2. Sequence

3.2.1. User selects the Indie Game Showcase

3.2.2. System displays a wide range of indie games

3.3. There should be a playable demo that the user can download if the user wants to test the game.

3.4. Sequence

3.4.1. User selects an indie game of their choosing

3.4.2. System displays a store page of the game as well as a download link for a demo of the game

3.5. If the user likes the game, then they should be able to give a rating based on what qualities they enjoy about the game and/or donate to the developers to help aid them in development.

3.6. Sequence

3.6.1. User finishes playing the indie game

3.6.2. System will pop out a window that asks for a rating from 1 to 5 as well as a text box so that the user can praise or give criticism for a game

3.6.3. User fills out the review section

3.6.4. System will take the information and input it to the creators store page as well as send a copy of the review to the creators by email

3.6.5. If system does not detect information in the review section, it redirects the user back to the tab of their choosing.

4. Store

4.1. User should be able to access a storefront where indie and AAA games are available to buy.

4.2. Sequence

4.2.1. User selects the marketplace tab

4.2.2. System transports them to a marketplace where users can buy indie games as well as AAA games

4.3. There should be a feature page that primarily shows indie titles and when clicked transports the User to the featured games' page

4.4. Sequence

4.4.1. User selects the marketplace tab

4.4.2. System displays a self-rotating mini page in which indie games that are the most popular and/or sponsored gain attention

5. Community

5.1. Users can engage in the community page, where they can engage in video game news and create threads.

5.2. Community Sequence

5.2.1. User selects the community page

5.2.2. User shall have the choice between the news or threads subpage

5.2.3. User engages content and scrolls down to the comments

5.2.4. User creates a comment and sends

5.2.5. System records the information and posts the comment on the news board/thread

5.3. Developers can create update logs on their games.

5.4. Developer Sequence

5.4.1. Developer creates an update log

5.4.2. System creates a page to which the developer can customize to their liking.

List of Non-Functional Specifications

Performance Requirements:

1. ***Responsiveness***: Operates on various monitor sizes, ranging from 10” laptops to 38” desktop monitors. It will also be responsive with a wide variety of resolutions, from 1024 x 600 through 3840 x 2160.

2. ***Reliability:*** The system should be operational 100% of the time, unless scheduled maintenance is being performed. YES

3. ***Storage Utilization:*** Storage should be considered heavily, and the base platform should have as small a size as possible to allow multiple downloads of games for each user without impacting their storage requirements. YES

Ease of Use:

1. ***Training Time:*** Program shall be clearly and intuitively usable post installation, with no need for extraneous “tips” or “tutorials”. YES

Interoperability Requirements:

1. ***Computer and OS Compatibility:*** The system will operate on the most modern Windows OS and one generation previous. Compatibility for other OSes will be considered for the future. YES

Expected Load:

1. Expected load will be very high, as the program will ideally be used by as many people as possible concurrently to foster a community environment. YES

Security Requirements:

1. ***Login/Password System:*** A login/password system to maintain the preferences, ratings and downloads of all users. YES (ADD HASH FUNCTION)

2. ***Encryption:*** A third-party application will be used to handle any monetary transactions. ??

Portability Requirements:

1. ***Platform Compatibility:*** This system will work on any portable device that runs Windows OS as long as it is a standard desktop version of Windows OS installed. YES

Storage Requirements:

1. The storage for our system will be maintained presently on GitHub and a hosting site in the future. YES

Availability Requirements:

1. ***Accessible Times:*** Our system should be available for use 24 hours a day, 7 days a week. It will be up and running as long as the server is available. YES

2. ***Support:*** There will be support availability by email. They will be responsive within 72 hours. YES

High-Level System Architecture and Database Organization

1. **Github/Github Desktop:** Will be used for keeping track of work to be done, what is being worked on, and what has been completed. It will also be used for downloading the current build of the software and updating parts of the current build through its branch feature.

2. **Discord:** Will be used as the team's main source of communication between each other. It will be used for personal communication, asking members for assistance, and the meeting place for scrums with voice and video call capabilities.

3. **VSCode:** The IDE/Code editor that will be used by our development team.
    a. C++: will be used for backend development
    b. JavaScript/CSS/HTML: will be used for client side development

4. **Django REST:** A web development framework that uses python. Will be used for backend development.

5. **Django's Default Database:** Will be used for the database of this project in the early stages.

6. **MySQL:** Will be used for the database of this project.

7. **Electron JS:** Will be the framework used for desktop app development using JavaScript, CSS, and HTML.

8. **Adobe Illustrator/Photoshop:** Will be used for creating UI/GUI elements and front end development.

9. **Godot:** Game engine that is used for some of the sample games that will be displayed in our project.

10. **DB Organization:** Our database will have tables for holding our users' account information (including their owned/created software/games), for holding every game in our store page, for chat logs, and for holding community posts/threads. The account table will hold all user information from login credentials to the communities they're a part of.

11. **Media Storage:** Media storage will be kept in the database. As for format requirements, nothing specific has been laid out yet. I would predict we would stick to the more common formats: png, jpeg, mp3, mp4, gif, etc.

12. **Search/filter architecture and implementation:** The algorithm for search that will be used is a form of binary search. Accounts, software on the discover page, chats, and community threads will all be searched for either sorted alphabetically or via a tag system. Some form of pointer system will likely be used to access data associated with locally stored files (for instance, a pointer to a cloud save of a user owned game).

13. **Our own APIs:** These can be seen in the UML class diagram

    a. **User/Account:** These API's deal with user account info stored in the database and used for logging in. They are used as a median to assist other API's in getting account specific information.

    b. **Library:** A client side API that accesses a user's games held in local storage and launches them.

    c. **Create:** An API that allows for a user to upload and manage their created software.

    d. **Discover: An** API that allows users to explore our game/software catalog and purchase sofware..

    e. **Community:** An API that allows users to explore community pages for their favorite games and software as well as letting the post to community boards.

    f. **Profile:** An API that allows a user to view and change aspects of their account.

    g. **Settings:** An API that lets a user change the settings for their client.

14. **Describe any significant non-trivial algorithm or process:** The only significant non-search process envisioned so far is the main purpose of our software, downloading and launching games on/from users' local storage.

Competitive Analysis

This section will analyze major competitors of Lynx in the digital distribution service and storefront for video games and software. The main competitors of Lynx will be Steam, itch.io, GOG.com, Humble Store, and Epic Games. This analysis will focus on 6 major features provided by each competitor and Lynx (Store Page, User Experience and Interface, Community Features, Support for Developers, Revenue Model, Data Privacy) and three additional features (Indie Game Focus and Variety, User Recommendations, Cosmetics). Using a numerical scale (1=bad, 2=poor, 3=fair, 4=good, 5=outstanding) for each feature, Lynx will be compared against its top five competitors.

	Lynx	**Steam**	**Itch.io**	**GOG.com**	**Humble Store**	**Epic Games**
Store Page:	5	5	4	3	4	3
User Experience & Interface:	4	4	2	3	3	3
Community Features:	5	3	3	3	1	2
Support for Developers:	5	3	5	3	4	3
Revenue Model:	4	3	2	3	3	3
Data Privacy:	5	5	5	4	5	2
Indie Game Focus & Variety:	5	3	5	2	2	2
User Recommendations:	5	4	3	3	4	3
Cosmetics:	5	3	1	2	1	2
Mean:	**4.75**	**4.13**	**3.75**	**3.25**	**3.38**	**2.88**

Lynx (4.75) https://lynxgamestore.com/

The Lynx game store page will be designed to be both attractive and user-friendly, showcasing a broad range of games while also efficiently organizing these offerings based on user preferences. The Lynx PC software will facilitate easy navigation to the store page, game library, community section, and user profile pages. The community section will host game forums, fundraising initiatives for upcoming indie games, and dedicated discussion and Q&A areas for individual games and software. Lynx will prioritize supporting developers with various tools and revenue opportunities, taking only a modest 10% cut of game profits. The primary revenue model for Lynx will revolve around the sale of engaging cosmetics and digital items that can be utilized in community discussions, friend communications, and during gameplay. Upholding data privacy and encrypting user data will be paramount to prevent data breaches and the sale of sensitive user information. Promoting a vast array of indie games to users will be Lynx's top priority, ensuring a substantial selection of titles are available to consumers and categorizing them based on user preferences and purchase history using an advanced algorithm.

Steam (4.13) https://store.steampowered.com/

Steam has carved out a niche as one of the most-favored digital storefronts for computer games and software worldwide due to its organized interface and extensive collection of titles. From blockbuster AAA classics to indie gems available in-store, where there are about roughly 58,000 indie game choices on this platform. Despite this figure showing an impressive number - itch.io has almost 800,000 smaller team-created masterpiece titles that showcase artistic creativity at its finest! While Steam does provide some support for creators and partners' communities; several features like fundraising opportunities intended toward upcoming projects lack adequate representation compared with other competitors. The facade created by cosmetic items displayed on Steam can be deceiving, considering they offer limited usability or gradually expire after seasonal events conclude! Unfortunately - Steam takes a significant financial cut (30%) from developers which sparks criticism among most since they give-up too much of their shared revenue, hence limiting business strategies intended towards future growth.

Itch.io (3.75) https://itch.io/

With nearly 800,000 products created by smaller development teams, Itch.io is a vast digital marketplace catering to content creators and independent game developers alike. Its revenue model only takes a 10% cut from sales - a considerable advantage that attracts indie developers searching for profitable channels. Although games can be downloaded through its app directly onto users' computers, it offers fewer features than players enjoy on platforms like Steam's client; consequently making engagement in community activities or forums quite strenuous without toggling between the app and browser tabs. Sadly, there are only a few user profile customization possibilities along with cosmetic preferences and social functions while utilizing itch.io. As such, it primarily shows off indie games exclusively, rather than exceptional features that would drive gamers to use their platform or app long-term.

GOG.com (3.25) https://www.gog.com/

For gamers seeking digital storefronts with a focus on independent game titles, GOG.com presents itself as one option among many available in the market. However, its product offering stands at roughly 2,500 indie games—a lower number than alternative platforms—and comes with the downside of taking a sizable percentage (30%) from developers' earnings. The site also delivers fewer customization features and cosmetic options than others do, while sporting only forum based community interaction channels as part of its offering repertoire. One positive quality worth noting is that it provides DRM free games suitable for playing offline—an advantage over competitors—but might not have broad enough using potential to warrant significant attention when considering overall game selection.

Humble Store (3.38) https://www.humblebundle.com/store

The Humble Store is a digital video game storefront which offers discounted game and software bundles while giving back through supporting charities. Even though they have a decent selection of big name titles like AAA games available at their virtual storefronts; consumers who browse the site will notice considerably fewer indie offerings by contrast— around 5,000 —than expected given how unconditionally supportive this platform seems towards uplifting newer

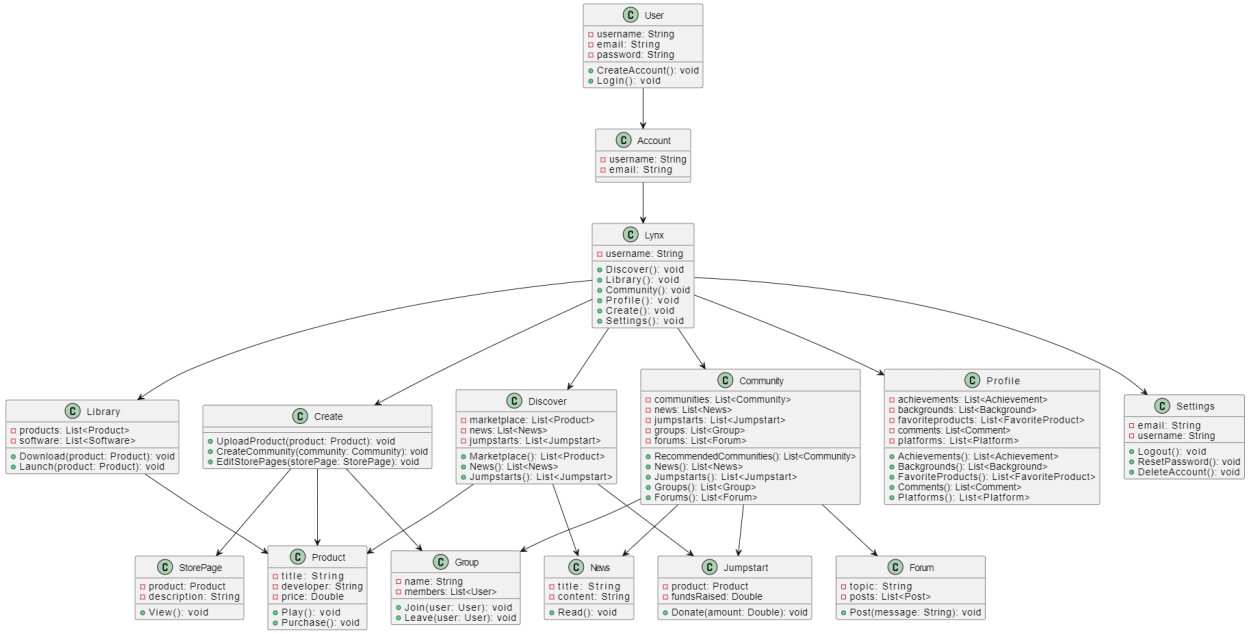
creators. The developers who want to utilize their publishing service may feel dissuaded as well since they charge a 25% fee. Additionally, the Humble Store has shortcomings when it comes to features like customization options, community features, and cosmetic items.

Epic Games (2.88) https://store.epicgames.com/

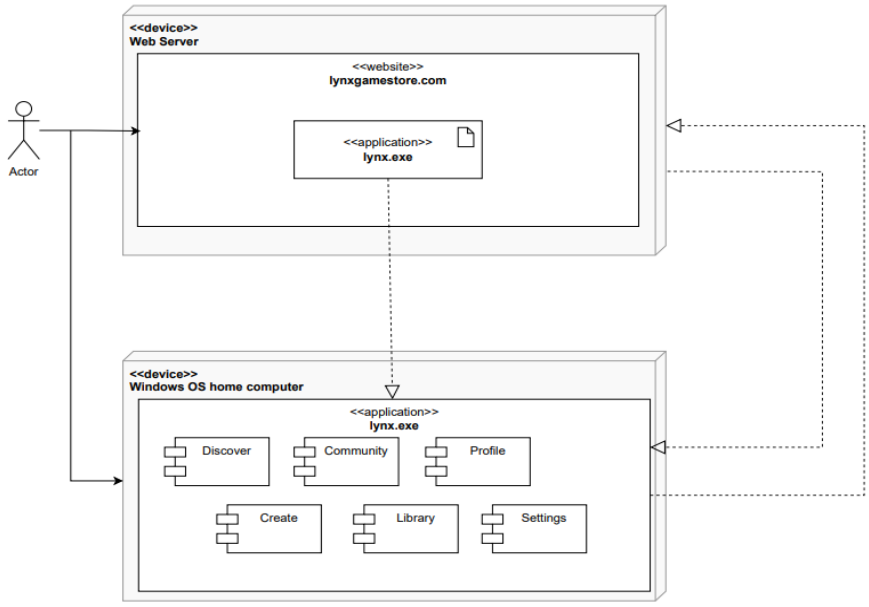
Exclusive titles make the Epic Games Store stand out amongst all other online marketplaces catering towards gamers. They further sweeten their offerings by taking only a mere 12% cut from developer earnings, making them an attractive option for creators wanting to sell their wares. Even so, compared to alternative digital marketplaces providing greater variety in terms of gaming experiences along with better community functionalities, the Epic Games Store has limitations. These also extend to customization options, given that user profile pages are somewhat basic as well. An event as severe as a data breach that Epic Games had in 2016, where over 250k users' details were compromised including email addresses, passwords and usernames, calls into question the measures they have in place when it comes to user data safety.

High-Level UML Diagrams

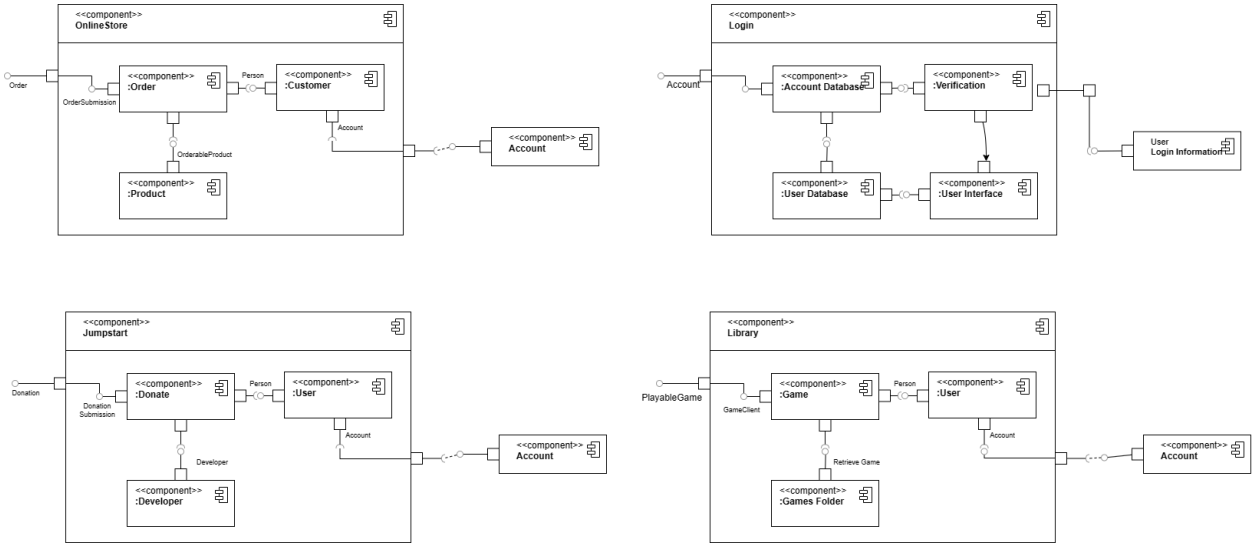
UML Class Diagram:



UML Class Diagram:



UML Class Diagram:



Key Risks for the Project

Skill Risks

- We have a lack of backend experience, so there may be some inconsistencies with the project in terms of integrating the systems to the backend. Frontend skills are also lacking, but we have been able to create workarounds for those problems.

Schedule Risks

- Ideally, creating Lynx would require many months to complete, to which we would not have enough time to fully develop the project. Deadlines were met, and every member has an open schedule. The team has also made consistent and progressive progress.

Technical Risks

- There are some slight inconsistencies on the database, so account creation may not work as intended. Workarounds have been developed and are working towards more reliable solutions.

Teamwork Risks

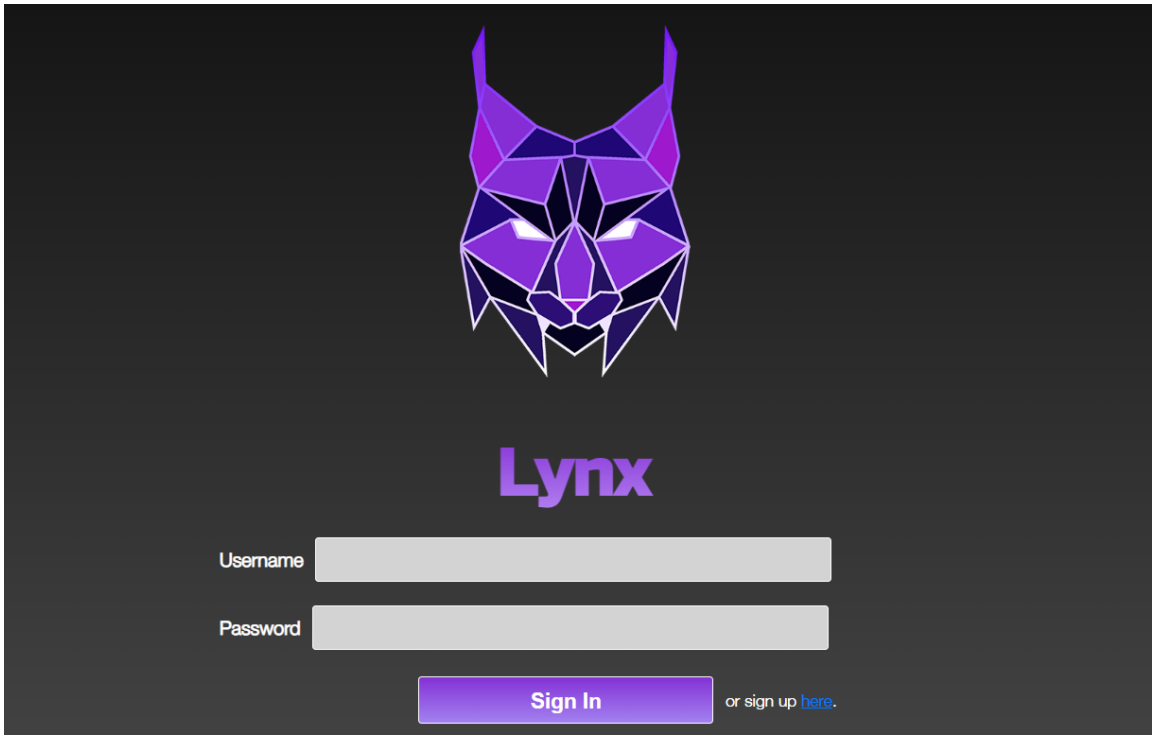
- In terms of teamwork, there are no issues regarding disagreements among the team. Everyone in the team is able to perform their duties well. Some team members needed help with their development, to which other team members were able to help them to the best of their ability.

Legal/Content Risks

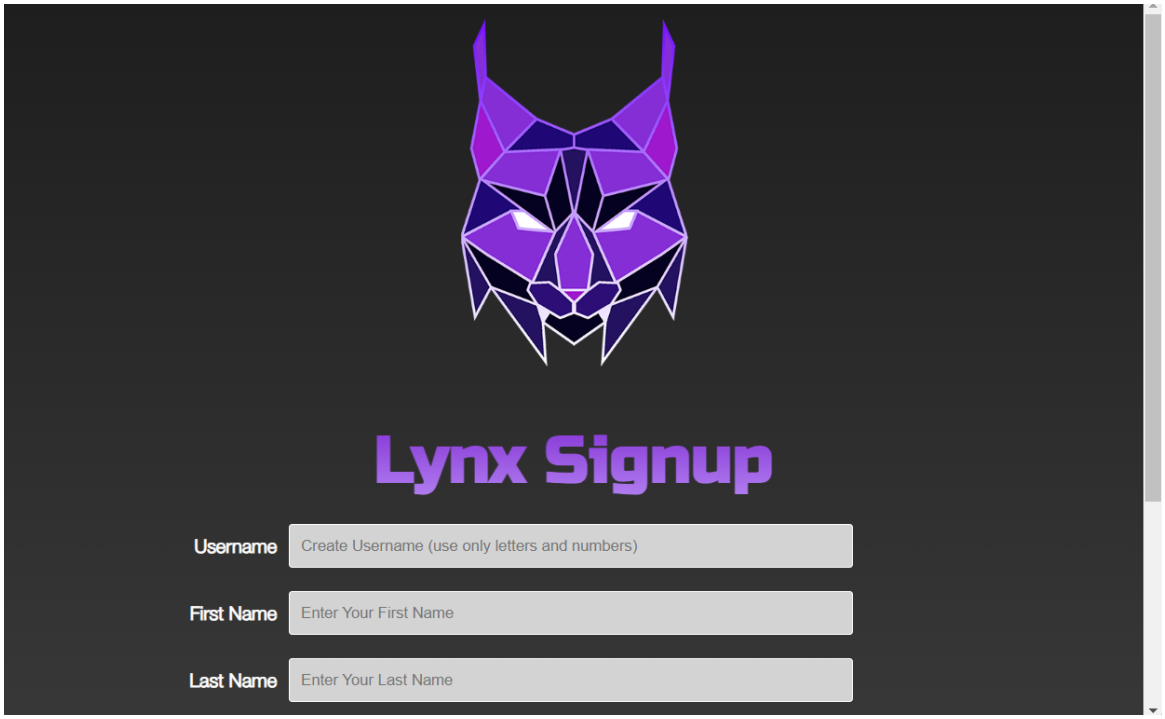
- In terms of legal risks, we have not established proper licensing/copyright, but we are implementing them in the future. In terms of content, all forms of content are obtained legally.

Screenshots of Lynx

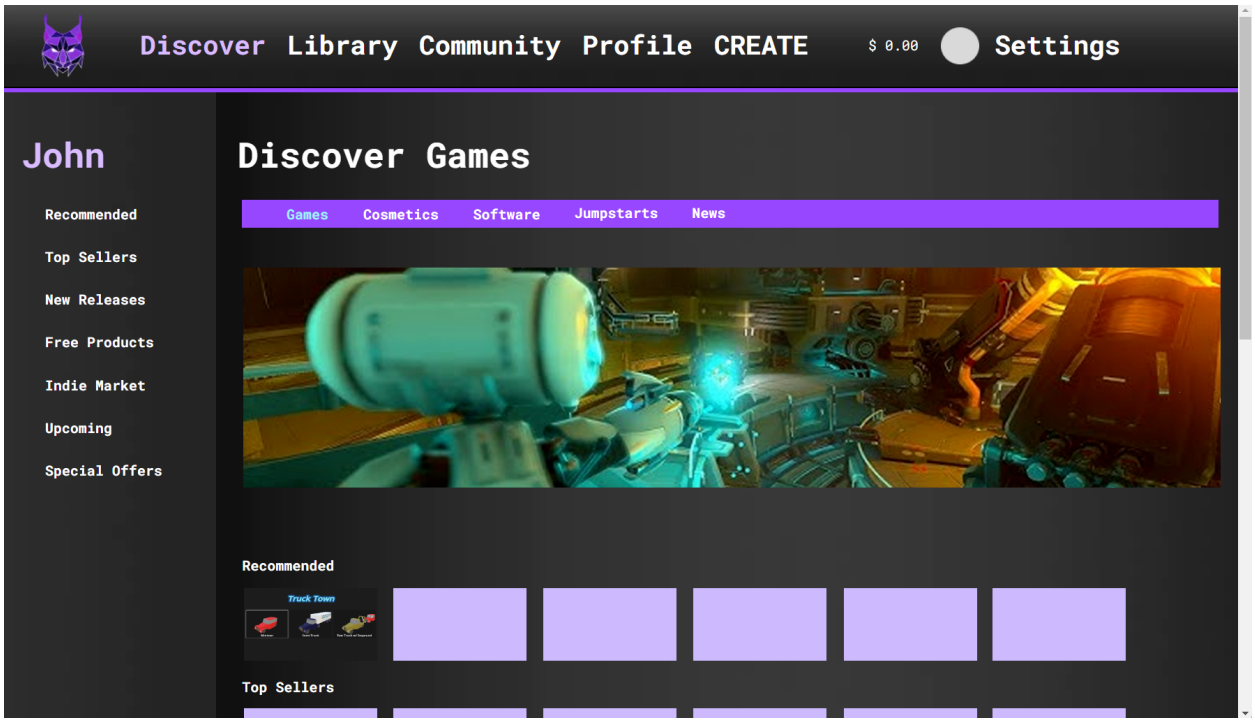
Login



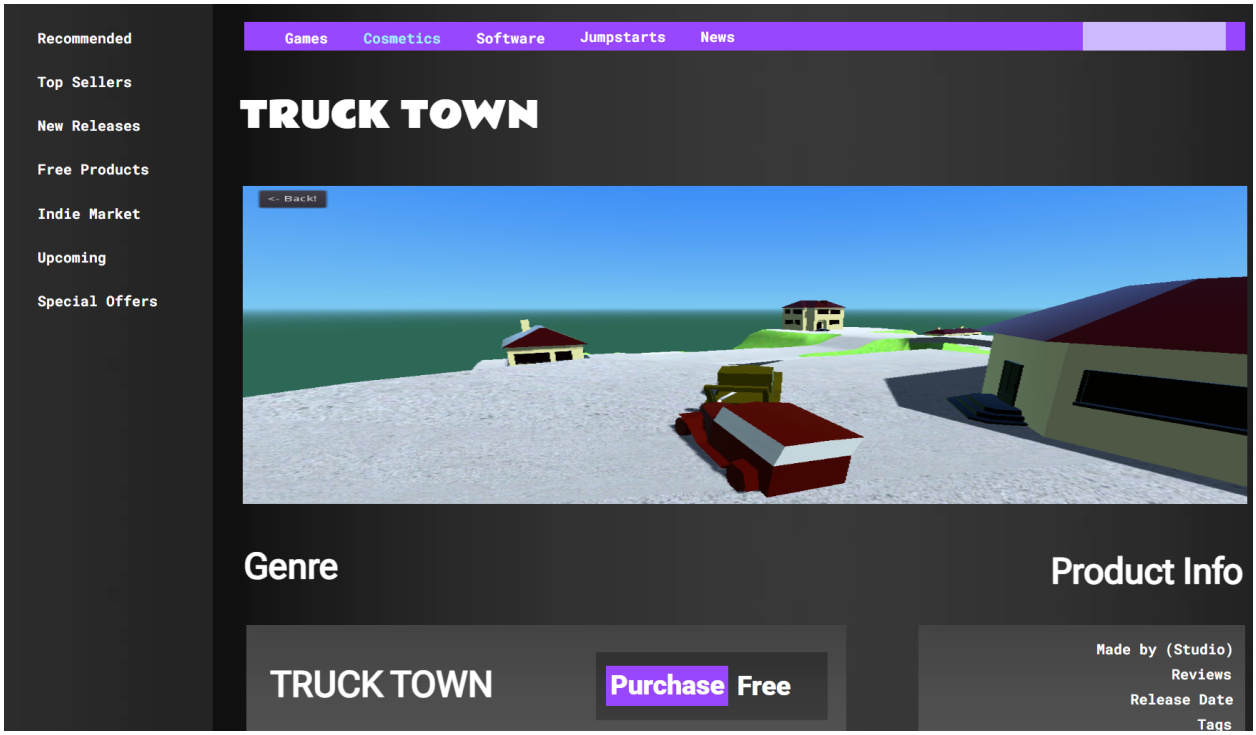
Sign Up



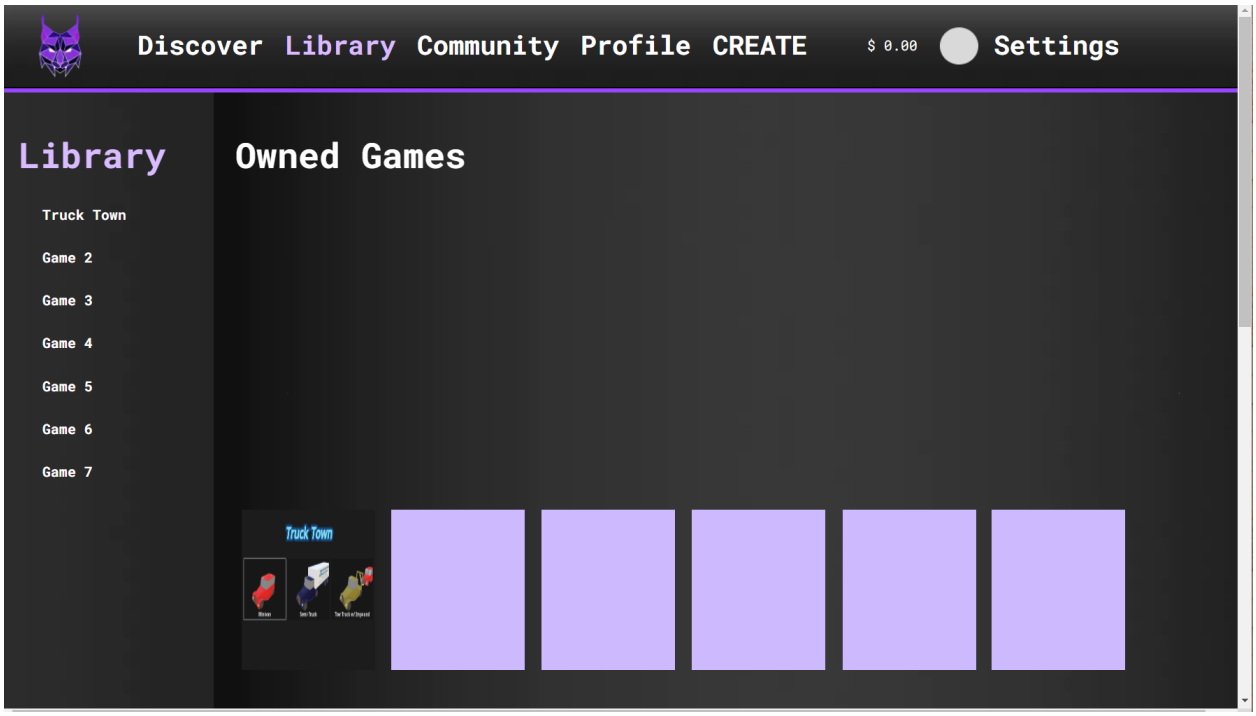
Discover Games Page



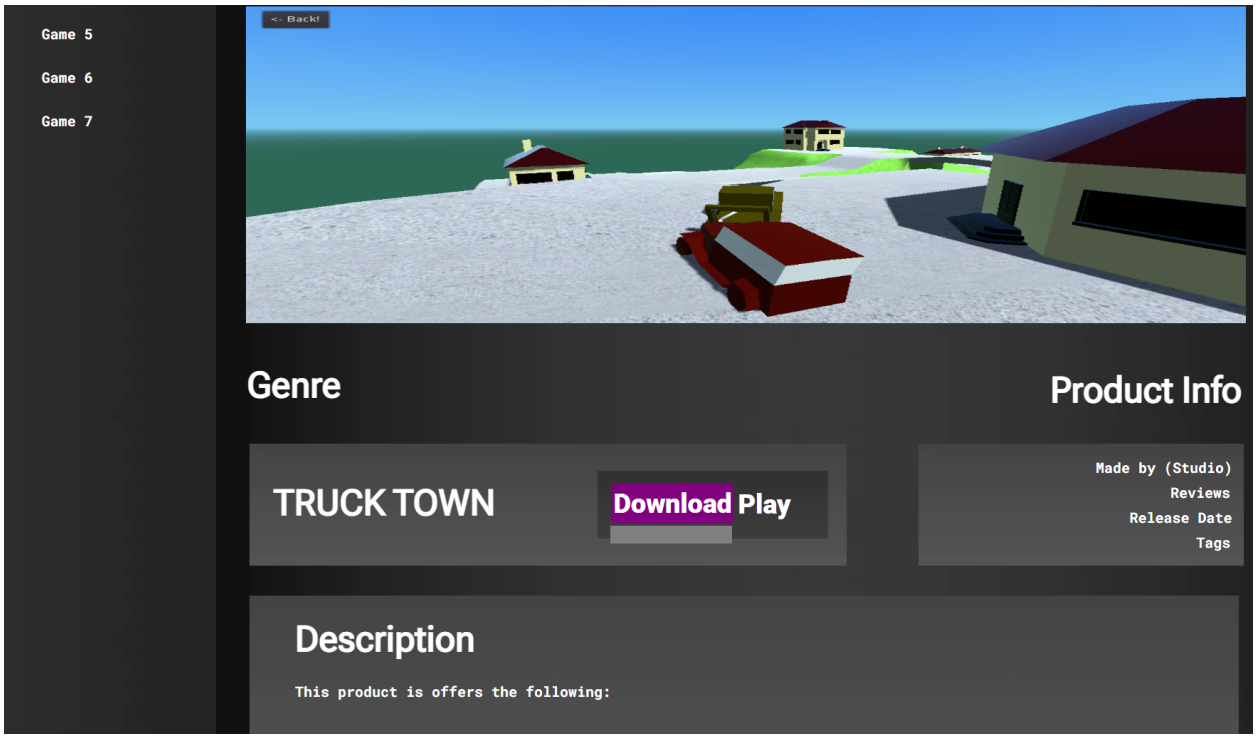
Product Page



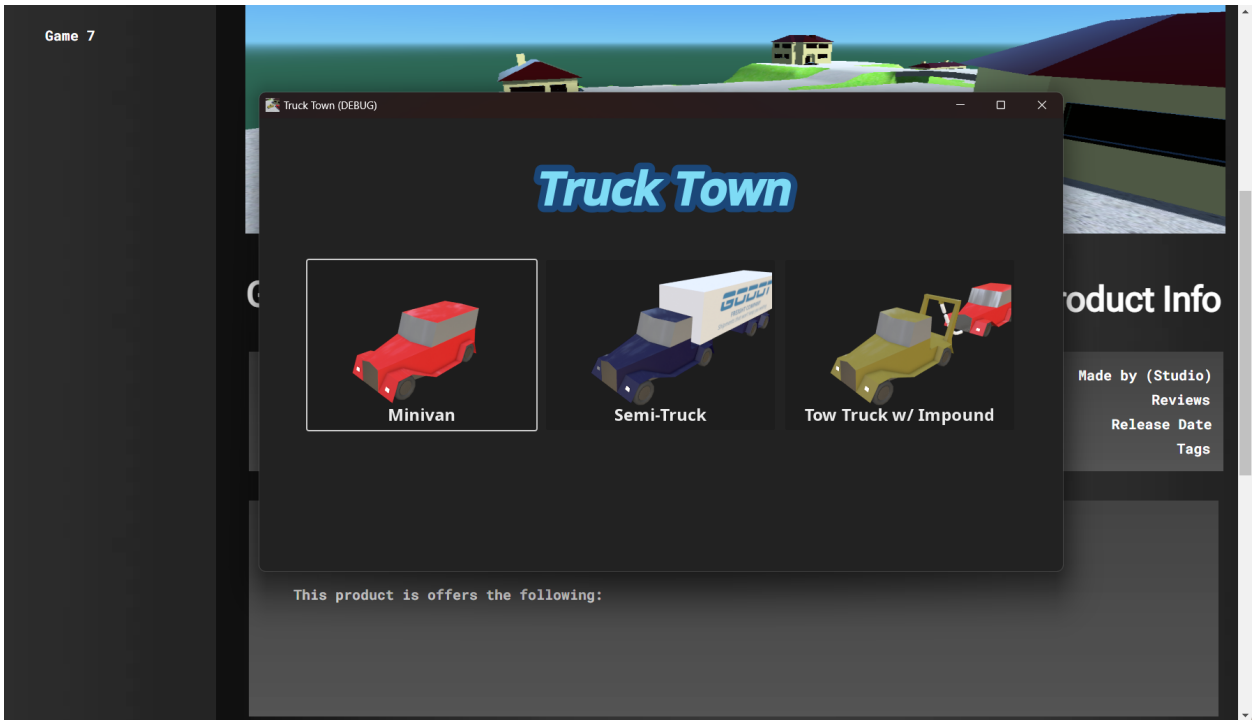
Library Page



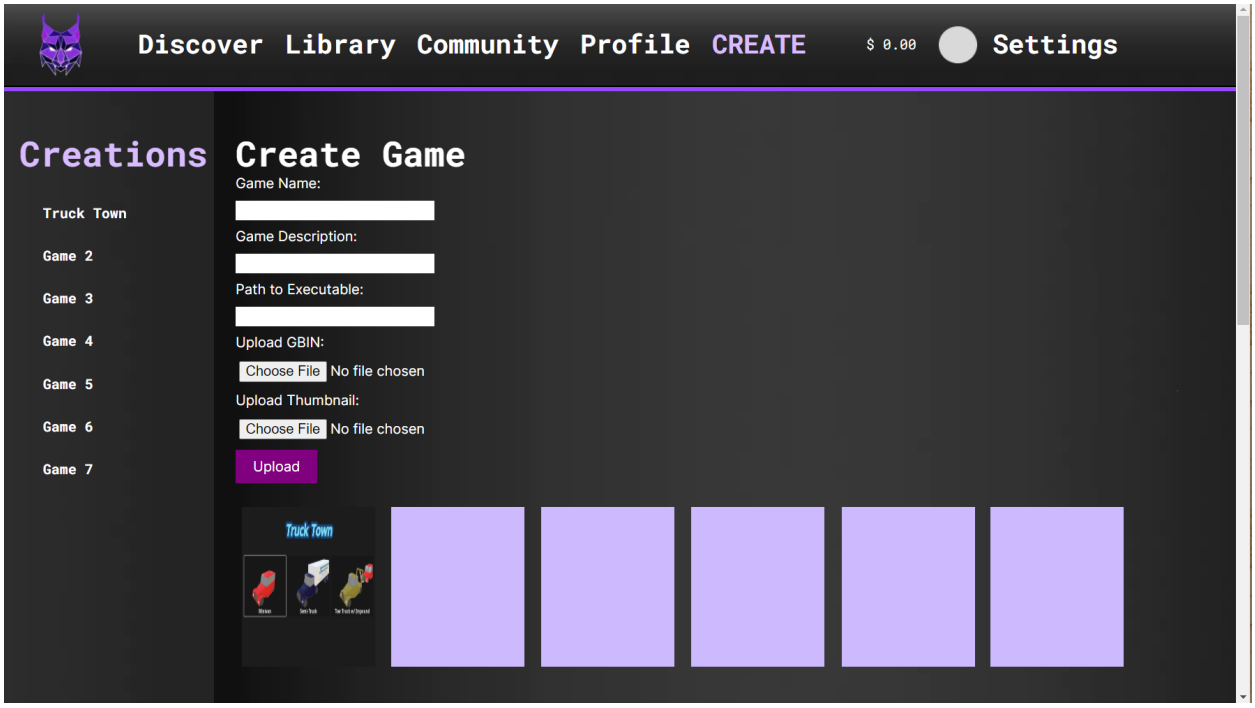
Game Launch Page



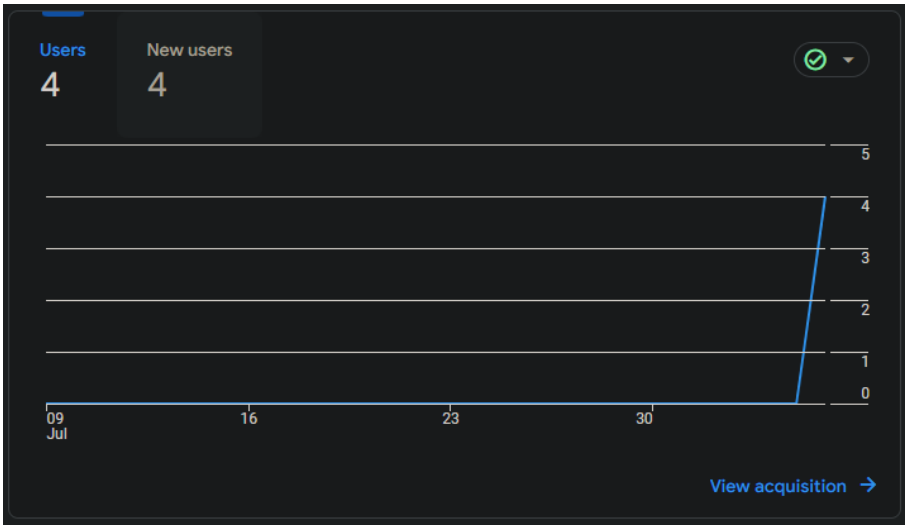
Game Launching

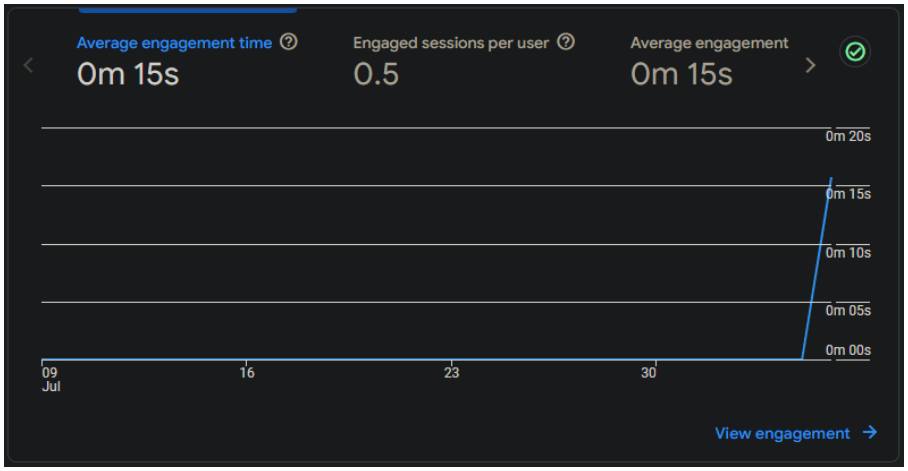


CREATE tab (upload functionality)



Google Analytics





Above are two of the plots from our website's, lynxgamestore.com, google analytics page. There is a plot of our total users and new users along with our average engagement time per user; both of which graph their respective data over a 4 week period. There are other useful plots and data to be found using google's analytics, including realtime data and monetization data. It even categorizes users of the site by their demographics. However, there is a distinct lack of data collected so far in the above plots. Since our software is a desktop application, the website hasn't been used in development. If the Lynx team continues this project in the future, we hope to implement google analytics into the client, if possible.

Team Member Contribution

Mike Clopton (20/20) - Mike Clopton is the Product Owner of the project, planning the documentation and legalities of the project.

- 10 submissions

Rodolfo Andres Rivas Matta (20/20) - Rodolfo Andres Rivas Matta is the Team Lead of the project, appointing and managing the responsibilities of the team members of the project.

- 10 submissions

Hunter Bresler (20/20) - Hunter Bresler was the Back-end developer/lead of the project, working on API integration and made final decisions on the features implemented on the project.

- 10 submissions

Cody Smith (20/20) - Cody Smith was the Front-end developer of the project, working on the UI functionality of the project.

- 10 submissions

Normil Luccin (20/20) - Normil Luccin oversaw all contributions and commitments on Github as the Github master. Also assisted in the overall documentation of the Lynx project.

- 10 submissions

Post-Project Analysis

Main Challenges:

- Implementation of APIs was difficult to implement
- Time management
- Determining overall scope of the project and setting realistic goals

Main Takeaways:

- To address the API problem, we made temporary workarounds on the features that utilized the API and created valid solutions later on throughout development.
- For resolving the time management, we set soft deadlines to team member assignments according to their schedule, which resulted in a better development cycle and minimal work errors.
- In terms of our overall scope, we decided as a team to cut back on most of the features and focus on high-priority features that us as a team can implement efficiently. Due to the restricted timeframe, we instead try to set and complete small tasks throughout each milestone, which resulted in much better workflow.