Table of Contents

[1 Abstract 2](#_Toc149826759)

[2 Introduction 2](#_Toc149826760)

[3 Literature Review 2](#_Toc149826761)

[3.1 Review of Game Portal 2](#_Toc149826762)

[3.2 Differences between technologies 5](#_Toc149826763)

[3.2.1 Front-End 5](#_Toc149826764)

[3.2.2 Back-End 7](#_Toc149826765)

[3.2.3 API 7](#_Toc149826766)

[3.2.4 Database 8](#_Toc149826767)

[3.3 Review of Methodologies 8](#_Toc149826768)

[3.4 Choosing solutions 8](#_Toc149826769)

[4 Requirement Analysis 9](#_Toc149826770)

[4.1 Similar application 1 9](#_Toc149826771)

[4.2 Similar application 2 9](#_Toc149826772)

[4.3 Conclusion 9](#_Toc149826773)

[5 Software design 9](#_Toc149826774)

[5.1 Architecture (optional) 9](#_Toc149826775)

[5.2 GUI (optional) 9](#_Toc149826776)

[5.3 DB (optional) 9](#_Toc149826777)

[5.4 UML (optional) 9](#_Toc149826778)

[6 Software implementation 9](#_Toc149826779)

[6.1 Development environment 9](#_Toc149826780)

[6.2 Important technical problems & solutions 9](#_Toc149826781)

[6.3 Test (optional) 9](#_Toc149826782)

[6.4 Results 9](#_Toc149826783)

[7 Evaluation and conclusion 9](#_Toc149826784)

[7.1 Evaluation of results 9](#_Toc149826785)

[7.2 Conclusion 9](#_Toc149826786)

[8 Appendix 9](#_Toc149826787)

# Abstract

* Summary of the whole report

# Introduction

* Introduction to the project

# Literature Review

## Review of Game Portal

* **Fix:**
  + Future of the domain
  + Thêm các bảng dữ liệu

Video games have always been a favorite type of entertainment since the beginning of the 1970s when the first games “Pong” and “Pac-Man” arrived. From the beginning, “Pong” and “Pac-Man” were only available in arcade machines since it was in the early state of technology.

As time keeps flowing, the evolution of technologies and the interest of people in video games keeps growing. More and more video games were made and people wanted to share their ideas and video games with everyone around the world.

In the early 1990s, America Online (AOL) and CompuServe were the first online services that provided platforms for multiplayer gaming and game downloads and they are the original form of game portal. According to Hume(1995), the width, depth, and friendliness of AOL and CompuServe are opposite, both companies offered Email, newswires, publications, computer software, online "chat" groups, gaming, study resources, shopping, and Internet access at no cost. But at the time, this type of service was not yet popular and the definition of game portal was not clear since besides just game, they open a lot of different categories for different purposes.



Figure . CompuServe website ([L'Histoire de CompuServe (astrosurf.com)](http://www.astrosurf.com/luxorion/compuserve-histoire.htm))

In 1995, the Total Entertainment Network (TEN) was created via the combination of Planet Optigon and Outland, the first for-profit game network on the Internet. At the time, for access to TEN's collection of games, both exclusive and nonexclusive—most of which were PC games modified for online play—users had to pay anything from $10 to $30 per month. Instead of acting as a platform for game distribution—where users could buy and download titles—these services were primarily concerned with offering multiplayer features for already-existing games.

A screenshot of a video game

Description automatically generated

Figure . TEN website ([TEN Home](http://ten.net/199706/html/ten_home.html))

On September 12, 2003, Valve Corporation introduced to the world Steam, the next generation of game portal. According to Dunn(2013), the primary goal of Steam was to make it easier to provide patches and other updates for online games, especially “Counter-Strike” and “Half-Life”. Half-Life 2's release is largely responsible for Steam's current state, as seen by the game's domination of the storefront. Not that Valve wanted you to forget: even if you purchased the critically acclaimed shooter in a box at the shop, you still needed to register for Steam to play it. It was a turning point in the evolution of digital ownership from the previous paradigm, which used CDs and CD keys, to the current paradigm, which links game licenses to online accounts.

A screenshot of a video game

Description automatically generated

Figure . Steam ([Welcome to Steam (steampowered.com)](https://store.steampowered.com/)

If Steam was popular with PC players, the Xbox Games Store and PlayStation Network are the two most famous game portals for Xbox Console and PlayStation Console. With the introduction of Xbox 360 in 2005, Microsoft created a subscription service and integrated it into the Xbox system, instead of using memory cards like the older version. In late 2017, Microsoft replaced the Xbox Games Store with the Microsoft Store on the Xbox One.

PlayStation Network (PSN) launched in 2006 and it was originally for the PlayStation game console but soon extended to smartphones, tablets, etc. The PlayStation 3 was the first one to apply PSN to the system. In 2010, Sony gave players an optional premium subscription service on top of the PSN service that was already free. The subscription is called “PlayStation Plus” and it provides players with exclusive content, games early access, and discounts. For security reasons, the PSN has been shut down for two months since April 2011. Since then, PSN has been the most famous game portal for consoles, and with the availability of the exclusive game “Spider-man” on PlayStation 5, they have confirmed that there are no opponents since the downfall of the Xbox Games Store.

Unlike Steam or Epic Games Store where they provide a huge selection of games from several publishers and studios, smaller game portal like Riot Games offers access to a variety of games, community features, and updates, its main concentration is on games created or released by Riot Games

## Review web technologies

Continuing with the above topic, this section will mention various technologies that could help me develop the game portal for “Land of Ringo”. These technologies will be divided into many sections that suit their main function, which are Front-end, Back-end, API, and Database. First, I will begin with Front-end.

### Front-End

* **Fix:** 
  + Thay các ngôn ngữ = framework
  + Nói về framework trước rồi tóm tắt lại = bảng

Front-end is the factor for users to choose to stay at a website or not since it affects them directly. Increased user happiness and engagement are the results of a well-thought-out and functional front end. On the other hand, a subpar user interface may cause consumers to become frustrated and quit the website or service. Here are some criteria that are essential for front-end development:

* Browser compatibility: Code might be interpreted differently by different web browsers. For a unified user experience, selecting a language that guarantees consistent behavior and appearance across all browsers is essential.
* Scalability: The front-end of your program must be able to manage growing loads and levels of complexity as it expands in both user base and features. Making a scalable technological selection will guarantee the seamless expansion of your application.
* Mobile responsiveness: Most website traffic originates from mobile devices. Selecting a technology that facilitates the creation of adaptable designs that function properly across all screen sizes is essential.
* Maintainability: It is simpler to update, debug, and extend maintainable code. A language or framework that promotes modularity, clean code, and best practices will help a project succeed in the long run.

|  |  |  |
| --- | --- | --- |
| **Front-end** | **Pros** | **Cons** |
| Jquery |  |  |
| Vue.js |  |  |
| Angular |  |  |
| React |  |  |
| Ember.js |  |  |

Table . Front-end programming language pros and cons

### Back-End

The back-end is where the server, database, and server-side apps are all located, along with the application. It oversees carrying out operations and computations, as well as handling and storing data. Data is sent and received between the front-end and back-end for display or manipulation by the user. Here are some criteria that are essential for back-end development:

* Performance and scalability: The performance requirements of the application should be met by the selected language, particularly for computation-intensive jobs. As the quantity of data or the number of users increases, it should likewise scale well.
* Maintainability: Clean, clear, and modular code is easier to maintain and update over time, lowering the likelihood of problems and technical debt.
* Compatibility and integration: The back-end language should work effectively with any external services or APIs you want to employ, as well as the other technologies in your stack.
* Cross-Platform Development: A programming language that facilitates cross-platform development may prove advantageous if your application must function across many platforms, such as the web, mobile, and desktop.

|  |  |  |
| --- | --- | --- |
| **Back-end programming language** | **Pros** | **Cons** |
| Django |  |  |
| ASP.NET |  |  |
| Express |  |  |
| Flask |  |  |

Table . Back-end programming language pros and cons

### Database

An electronic collection of organized data that can be accessed and stored is called a database. Databases are made to handle safely and effectively, and store, retrieve, and alter data. They are essential for many applications because they offer a means of storing data permanently and retrieving it when required.

* Performance: The user experience of your application is directly impacted by the database's performance. Improved reading and write speeds can result in faster load times and a more seamless end-user experience.
* Security: Security features guarantee that sensitive data is protected and that only systems or people with permission may access or change it. Upholding user trust and complying with regulations depend on this.
* Scalability: The database I select should be able to expand to meet the demands of your application. This might include managing greater amounts of data, more users at once, or both. For distributed systems, horizontal scalability—that is, adding more machines—is frequently chosen because it provides more robustness and flexibility.
* Consistency and data integrity: To keep users trusting the program, data consistency must be guaranteed across all actions and nodes (particularly in distributed databases). Constraints on data integrity aid in preventing incorrect data from entering the system.

|  |  |  |
| --- | --- | --- |
| **Database** | **Pros** | **Cons** |
| MongoDB |  |  |
| Firebase |  |  |
| dick |  |  |

Table .Databases pros and cons

## Review of Methodologies

* Review related methodologies
* Compare, pros / cons

|  |  |  |
| --- | --- | --- |
| **Methodologies** | **Pros** | **Cons** |
| Waterfall |  |  |
| Spiral |  |  |
| Agile |  |  |
| Scrum |  |  |
| Kanban |  |  |

## Choosing solutions

* Discussion of your chosen technologies / methodologies
* Front-end: Vue/Angular/Jquery
* Back-end: NodeJs
* Database: MongoDB
* Methodology: Agile/Spiral

# Requirement Analysis

## Similar application 1

* Riot games

## Similar application 2

* Steam

## Conclusion

* Confirm your application features
* What is your improvements / modification / localization / …?

# Software design

## Architecture (optional)

## GUI (optional)

## DB (optional)

## UML (optional)

# Software implementation

## Development environment

## Important technical problems & solutions

## Test (optional)

## Results

# Evaluation and conclusion

## Evaluation of results

* Pros / Cons of your application

## Conclusion

* Lessons learnt
* Problems / difficulties
* Future improvements
* Conclusion

# Appendix

* Final plan
* Screenshots (optional)