

**VIETNAM NATIONAL UNIVERSITY**

**UNIVERSITY OF INFORMATION**

**TECHNOLOGY INFORMATION SYSTEMS FACULTY**

**REPORT**

**COURSE: DATABASE SYSTEM AND DESIGN**

**LECTURER: Dr. Nguyen Thanh Binh**

**CLASS: MSIS4013.O21.CTTT**

**Lê Hữu Bách – 21521844**

**Lê Nguyễn Gia Hưng – 21520890**

Introduction:

Warframe is a type of MMORPG game that is about exploring, farming, hack-and-slashing with many more features. As most items in-game are farmable, they are also “tradeable”, which makes Warframe has something to be noticed by many players, especially those who seek for collecting the item with fast method or to gain funds from those.

Knowing that, our team has come up with the idea to create Warframe Trading Hub - a trading website, for Warframe players to sell their items, or to find the one they need.

Scope

In this project, we are aiming to create a web-based trading platform for people in need of posting their rare items (Prime Parts, mods, etc...) to trade with other players ingame. By looking at those listed item in the main page, user can click on that to navigate to the trading hub of that specific merch, then they can proceed to trade with the person they want. After choosing the potential trader, user will be connected to that trader in the game. The chosen trader will be notified when there are a trade started in the game chat section.

There are serveral aspects that we have developed: first is the home page, which will show a list of in-game items that are added to the game’s database, update every patch. To organize things easier, we also decided to have a separated sections for each category. To create a trade for each item, we have a button that splits into 2 specific tasks: buy and sell, if a user post a “Want to buy” request, they will be asked to insert the item’s name, their price that they want to spend for the item and the quantity, after that the info will be uploaded to the “BUY” list in the specific item’s page, the same goes for “Want to sell” request.

We are also developing a “Quick Buy” page. This will be used to find the top 5 recent trade requests (on both buy and sell) that matches the user’s requirements for the item.

Literature Review:

The fact that this game is out since 2011, so there are already tons of trading website for Warframe players on the net. The most notable trading website that is currently used by everyone is warframe.market. The aforementioned website has been providing such stable experience with various functions support users for item trading. However, they still lack some features to enhance user experience even further, such as Quick buy feature, which is useful for those who hunt for mods with fluctuate value. As having a quickbuy feature, the user is able to find the item with the fastest time, without looking for just 1 trade request inside tons of others. Moreover, warframe.market does not seem to provide a user-friendly categorical features, which is quite inconvenient to look for an item without being distracted by other stuffs. Acknowledging all above problems they may encounter, our project will prove to affix those fatal points, providing fast and reliable trading hub for every Tenno out there.

Method/Solution:

To build our game item trading website, we have leveraged the power of React.js on the front-end, Node.js on the back-end, and MySQL for database management. This combination of technologies allows us to create a seamless and interactive trading platform for our users.

React.js, a popular JavaScript library for building user interfaces, forms the foundation of our website's front-end. With React.js, we have developed a rich and responsive user interface that enables users to browse available game items, create trade offers, and interact with other traders. The component-based architecture of React.js provides modularity and reusability, making it easier to manage complex UI elements and ensure a smooth user experience.

On the back-end, we have employed the power of Node.js, a server-side JavaScript runtime, to handle the business logic and API operations of our trading system. Using the Express.js framework, we have built a robust RESTful API that enables users to perform essential trading actions, such as creating trade offers, accepting or rejecting offers, and completing transactions. Node.js allows for efficient handling of concurrent requests and provides a scalable foundation for our trading website.

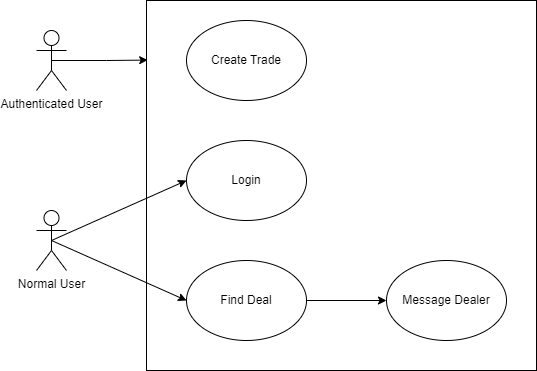
For database management, we have chosen MySQL, a widely used relational database management system. MySQL offers the reliability, performance, and flexibility required to store and manage our game item data. To interface with the MySQL database, we have utilized an ORM (Object-Relational Mapping) library, such as Sequelize. These ORM libraries provide an abstraction layer that simplifies database interactions, allowing us to define models, query the database, and handle relationships between entities using JavaScript code, without the need for writing raw SQL queries.

With MySQL, we have designed a well-structured database schema that accommodates the various entities of our trading system, including game items, user profiles, trade offers, and transaction records. By utilizing MySQL's features, such as indexing, transactions, and query optimization, we ensure efficient data retrieval and manipulation, enabling smooth and reliable trading operations.

In summary, our implementation of React.js, Node.js, and MySQL has provided us with a powerful and scalable foundation for our game item trading website. React.js facilitates an engaging user interface, Node.js handles the server-side logic and API operations, and MySQL ensures robust and efficient database management. Together, these technologies enable us to deliver a seamless and immersive trading experience for our users.

System design

Database:

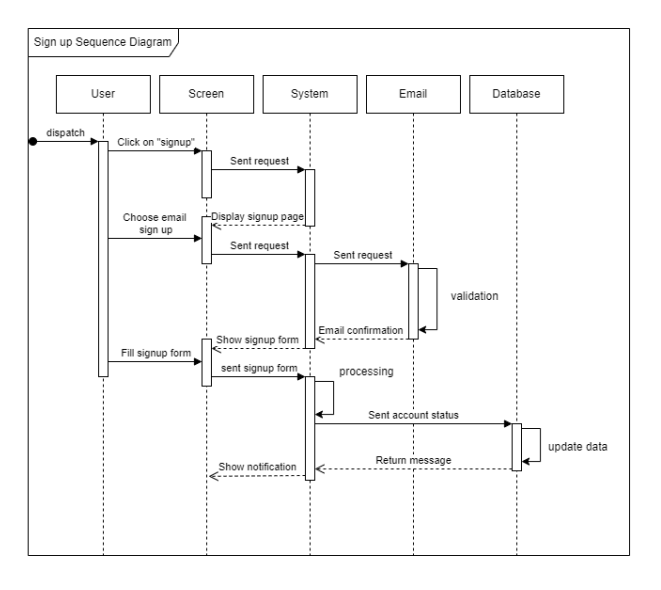
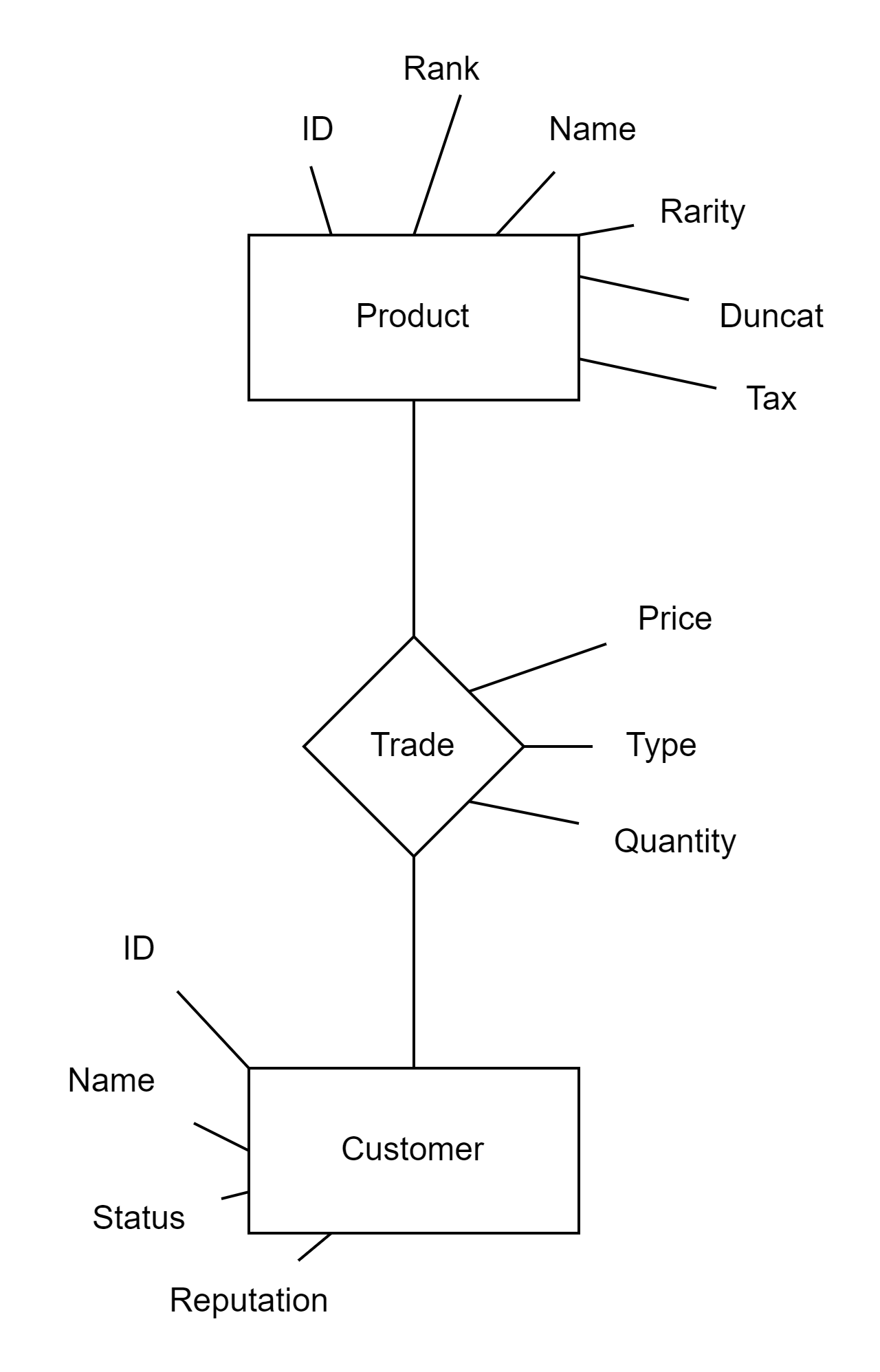


|  |  |
| --- | --- |
| Name | Create Trade |
| Description | As a user, I want to create a trade request in a website |
| Actor | Authenticated user |
| Trigger | When a user is authenticated then proceed to the trading section |
| Related Use case | None |
| Pre-condition | - User must be authenticated |
| Post-condition | - User’s trade requests are created completely and shown on the trading list  - Trade requests are saved into the system database |
| Flow of event | 1. User accesses to the website  2. User logs into the website  3. System shows trade create button  4. User clicks to choose to create a buy or sell request  5. User chooses item in dropdown list, input quantity and wanted price  6. User confirms  7. System uploads the request onto the list of the item. |
| Exception condition | If user hasn’t logged in, the button cannot be used |

|  |  |
| --- | --- |
| Name | Login |
| Description | As a user, I want to log into the website with my created account |
| Actor | Normal user, Authenticated user |
| Trigger | When a user wants to log into the website |
| Related Use case | None |
| Pre-condition | - User account must be existed and verified  - Internet connection required |
| Post-condition | - User is successfully logged in |
| Flow of event | 1. User accesses to the website  2. User clicks on the “Login” button on the top right corner  3. System shows login pop up form  4. User provides account information  5. System shows success login notification.  6. User status changes |
| Exception condition | If the user does not yet have an account, the system will ask to create.  If an account provided does not exist or wrong format, the system will show popup error. |

|  |  |
| --- | --- |
| Name | Find deal |
| Description | As a user, I want to look for fast deal for a specific item. |
| Actor | Normal user, Authenticated user |
| Trigger | When a user wants to look for deal in an item |
| Related Use case | Message dealer |
| Pre-condition | - User must access to the web  - Internet connection required |
| Post-condition | - User finds the needed deal |
| Flow of event | 1. User accesses to the homepage of the website  2. User clicks on the specific category  3. System shows list of items  4. User chooses a specific item in the item list  4. System shows lists of created trade requests with availability status |
| Exception condition | none |

|  |  |
| --- | --- |
| Name | Message dealer |
| Description | As a user, I want to notify the dealer whenever I want to trade with them |
| Actor | Normal user, Authenticated user |
| Trigger | When a user wants to notify other dealers |
| Related Use case | Find deal |
| Pre-condition | - Internet connection required |
| Post-condition | - the targeted dealer is notified |
| Flow of event | 1. User accesses to the website  2. User logged in  3. User chooses specific item in the item list  4. System shows lists of created trade requests with availability status  5. User clicks on a specific request  6. System will send a notification message to that chosen trader  7. User will be notified if that trader receives the message |
| Exception condition | None |



Conclusion

In conclusion, despite the existence of numerous trading websites for Warframe players, such as the well-known warframe.market, there are still areas where user experience can be further enhanced. Notably, the lack of a Quick Buy feature poses challenges for users seeking items with fluctuating values, as they must sift through numerous trade requests. Additionally, the absence of user-friendly categorical features makes it inconvenient for users to search for specific items without distractions.

In response to these limitations, our project aims to address these critical issues and establish a trading hub that offers fast and reliable services for all Tenno. By implementing a Quick Buy feature, users will be able to swiftly locate items and minimize the time spent searching through trade requests. Furthermore, our platform will prioritize a user-friendly interface with intuitive categorical features, ensuring a streamlined and efficient item search process.

Through these enhancements, our project seeks to provide Warframe players with an improved trading experience, offering a comprehensive solution that addresses the shortcomings of existing trading websites. By combining the stability and functionality of established platforms with innovative features, we aspire to create a trading hub that caters to the needs of every Tenno, fostering a vibrant and thriving trading community.

References

[warframe.market](https://warframe.market/)

[Index | Node.js v22.2.0 Documentation, n.d.](https://nodejs.org/docs/latest/api/)

[PedroTech. (2021, April 2). Full Stack Web Development Course](https://www.youtube.com/playlist?list=PLpPqplz6dKxUaZ630TY1BFIo5nP-_x-nL)

[*npm Docs*. (n.d.). https://docs.npmjs.com/](https://docs.npmjs.com/)