

Swinburne University of Technology

COS30049 - Computing Technology Innovation Project

Project Design Report

**Game Items Trading System**

**Group:**  2

**Assignment no:** 02

**Due date:** 10/11/2024

# I - Introduction

## 1. Project Description (Background and Motivation)

## The goal of the Game Items Trading System (GITS) project is to simplify and transform player-to-player game item trading. The gaming industry's explosive expansion and the growing need for a dependable and safe platform to handle these kinds of transactions are the driving forces behind this initiative. GITS will make advantage of state-of-the-art online technologies to offer a powerful database system, an intuitive user interface, and sophisticated search and filtering features.

## 2. Project Overview

#### **2.1. Objectives**

#### Make sure the UI is intuitive so that trading and exploring game stuff is a breeze. Provide a dependable and safe database system to hold item and transaction data. Use the extensive filtering and search options for effective item finding. Assure safe transactions and transfers of item ownership. Give users access to a clear and transparent transaction history.

#### **2.2. Scope**

##### The project scope outlines the goals and boundaries for the development of GITS strategically.

##### **2.2.1. In-scope**

##### Creation of an intuitive and aesthetically pleasing user interface. Building a solid and safe database system to hold information about items and transactions. Use of search and filter features to maximize the effectiveness of item searches. Guaranteeing trustworthy item ownership transfers and safe transactions. Creation of an intuitive user interface to facilitate transaction history access and evaluation.

##### **2.2.2. Out of scope**

Integration with particular gaming platforms or APIs.  
  
Creation of iOS and Android mobile applications.  
  
The use of cutting-edge trade tools like bidding or auction systems.  
  
Capabilities for community administration and extensive user support.

# II - Team Introduction

Our project team is made up of knowledgeable experts committed to seeing GITS through to completion. Every participant is essential to accomplishing the project's goals.

Team members:

**1. Vu Hoang Tung**

* Role: Project Manager

● Responsibilities:

○ Distributing tasks to team members

○ Asking questions on behalf of the group

○ Submitting deliverables

**2. Phan Duc Tuan**

* Role: Developer

● Responsibilities:

○ Developing and maintaining back-end functionalities

○ Ensuring seamless integration between front-end and back-end systems

**3. Nguyen Duy Anh Tu**

* Role: Developer, Tester
* Responsibilities:

○ Report maker

○ Collaborating with team to ensure the identification and resolution of defects

○ Verifying the functionality, performance, and security of the developed software

○ Producing thorough bug reports with clear documentation

# 

# III - Project Requirements

## 1. User Interface

## 

## The website must be flexible enough to adjust to different screen sizes. Every game object needs to be accessible to be seen and traded.

## This is achieved through responsive design using CSS in styles.css.

## The layout adapts to various screen sizes, ensuring accessibility on different devices.

## Game objects are displayed clearly in items.html with details

## 2. Database system

## 

## User profiles, transaction histories, and details on every game object should be kept in the database. Data security, accessibility, and integrity should all be guaranteed by the database.

## The database schema in ass2.sql defines tables for users, assets, and purchases.

## PHP scripts like db\_connect.php handle secure database connections.

## Data integrity is maintained through constraints (primary keys, foreign keys) in the database schema.

## 3. Search and Filter functionality

## 

## It should be possible for users to locate objects using the search bar. There should be different category-specific filters accessible.

## items.html includes a search bar.

## JavaScript in script.js handles dynamic filtering of items based on search input.

## The backend (search\_assets.php) supports searching by name or category.

## 4. Smart Contracts

## 

## Transferring item ownership and ensuring secure transactions are important.

## While not fully implemented in the provided code, the system architecture accounts for smart contracts.

## Smart contracts would handle secure item ownership transfers and transaction processing on a blockchain.

## 5. Transaction History

## It is necessary to save the transaction history, which shows previous deals. Transaction information must to be sorted and readily available.

# history.html displays the user's purchase history.

# JavaScript in script.js retrieves and displays purchase records from localStorage.

# The database schema includes a Purchases table to store transaction data.

# IV - Project Design

## 1. Overview

## With an emphasis on user empowerment, openness, and diversity, GITS seeks to establish a decentralized marketplace for the trade of game products. For both experienced traders and novices, the platform will offer an intuitive user interface that guarantees safe transactions and clear transfers of item ownership. With the goal of revolutionizing the seller-buyer dynamic in the gaming industry, GITS seeks to open up item trade to everybody while promoting constructive change in the industry.

## 2. System design architecture

## Strictly designed to integrate state-of-the-art technology, the system design architecture provides a smooth and safe decentralized game item trade platform. JavaScript, HTML, and CSS are used in front-end development to create a visually appealing user interface. The back-end prioritizes accessibility and integrity and uses technologies like Node.js and maybe a database like MySQL to manage data quickly. Robust systems, which may make use of blockchain technology or secure payment gateways, guarantee safe transactions.

## GITS employs a robust system architecture designed to provide a seamless and secure decentralized platform for trading game items.

## To provide traders with a dynamic and intuitive interface, the Game Items Trading System's front-end makes use of HTML, CSS, and JavaScript. The design places a high value on cross-browser compatibility, which enables users to access and utilize the platform without difficulty using any browser, and responsiveness, which guarantees compatibility across a range of screen sizes and devices.

## The system uses a MySQL database for dependable and effective data management on the back-end, as well as Node.js, a potent JavaScript runtime environment. With an emphasis on preserving data security and integrity, this combination guarantees the safe storage and retrieval of user data, item information, and transaction history.

## Additionally, the system uses smart contracts to enable safe transactions and user-to-user item ownership transfers. Despite not being completely implemented in the code that is supplied, these smart contracts are intended to improve platform trust and transparency by automating and securing the trading process.

## 

## 3. Database Design: The database schema is designed to efficiently store and manage user data, item information, and transaction history.

## 1. Users Table

|  |  |  |
| --- | --- | --- |
| Column | Data Type | Constraints |
| user\_id | INT | AUTO\_INCREMENT, PRIMARY KEY |
| username | VARCHAR(50) | NOT NULL, UNIQUE |
| password\_hash | VARCHAR(255) | NOT NULL |
| balance | DECIMAL(10, 2) | DEFAULT 100.00 |

## 2. Items Table

|  |  |  |
| --- | --- | --- |
| Column | Data Type | Constraints |
| id | INT | AUTO\_INCREMENT, PRIMARY KEY |
| name | VARCHAR(100) | NOT NULL |
| type | TEXT |  |
| price | DECIMAL(10, 2) | NOT NULL |

## 3. Purchases Table

|  |  |  |
| --- | --- | --- |
| Column | Data Type | Constraints |
| purchase\_id | INT | AUTO\_INCREMENT, PRIMARY KEY |
| buyer\_id | INT | FOREIGN KEY referencing Users(user\_id) |
| item\_id | INT | FOREIGN KEY referencing Items(id) |
| purchase\_time | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

## 4. API Documentation

## An essential component of the GITS project is the API (Application Programming Interface) documentation, which describes the interactions between the various software components in the system. With regard to GITS, the API documentation concentrates on the communication between the back-end (the database and server-side functionality) and front-end (what users see and interact with) of the website. The precise endpoints that enable this connection are described in the GITS API documentation. In essence, an endpoint is a URL that the front-end may utilize to make specific data requests or carry out certain operations.

## First off, the "Fetch Assets" endpoint is made to get details about every game object that is available from the database. The GET method, a common HTTP technique for data retrieval, is employed by this location. The back-end provides a JSON array of asset objects in response to a GET request sent by the front-end to this endpoint. A unique ID, name, description, category, price, and the ID of the owner are all included in each asset object, which represents a game item.

## Second, the "Search Assets" API makes it possible to look for certain assets in the database. This endpoint allows a request argument named keyword in addition to using the GET method. Users can look for assets by name or category with this keyword. This API returns a JSON array of asset objects in response to a GET request, same like the "Fetch Assets" service, except only those that match the supplied keyword are included.

## The system's overall efficiency and maintainability are enhanced by this methodical approach to API documentation, which guarantees uniformity and clarity in the communication between GITS's front-end and back-end.

## 5. Technical review

#### **5.1. Interface Design and Front-end Development**

* **Scope**: To direct the development and implementation of GITS's user interface, do a comprehensive technical investigation of pertinent online trading platforms with an emphasis on front-end development peculiarities.
* **Tasks**:

Examine the reference platforms' HTML structure, CSS style, and JavaScript interactions.  
  
 Optimize GITS's front-end development with best practices to enhance user experience.  
  
 Make sure it works with all browsers and is responsive.

#### **5.2. Back-end and Database development**

* **Scope**: Analyze the intricacies of database integration and back-end programming for GITS, making use of relevant technologies such as Node.js, Express.js, and maybe MySQL.
* **Tasks**:

Establish a solid process for back-end development using the relevant technologies.  
  
 Create and maintain a database to store data safely and effectively.  
  
 To safeguard user information, use security measures and make sure data integrity is maintained.

## 6. Website functions

#### The GITS website has 4 main subpages for different functions:

#### **6.1. Homepage**: Website homepage

A screenshot of a computer

Description automatically generated

There will probably be a hero section on this page with an attention-grabbing headline and call to action (like "View Items").  
 To encourage users, it could show off certain prominent or well-liked game goods.  
 It will be crucial to have easy access to the "Items," "History," and maybe even a "User Profile" area.

#### **6.2. Items:** Browse and search for game items

A screenshot of a game

Description automatically generated

#### This is where the HTML structure for the item list comes into play, represent an item with its details (name, type, price). JavaScript's ability to dynamically show and hide items depending on a search query will be used to construct the search bar. To further hone searches, think about introducing filter options (by item kind, for example).

#### **6.3. Purchase:** Confirmation of item purchase

A close-up of a website

Description automatically generated

Using JavaScript to obtain data, this page ("purchase.html") dynamically displays the name and price of the verified item.  
 The "View Purchase History" link is easily visible and takes users to their transaction histories.

#### 

#### **6.4. History:** User's purchase history

A grey and white website

Description automatically generated with medium confidence

The purchase records that are retrieved using JavaScript will be shown on "history.html," along with the item name and price of a previous transaction.

**6.5. General overview**

The website is a game item trading system where users can buy and sell virtual game items. It uses PHP for server-side scripting, HTML for the user interface, and a MySQL database to store data about users, items, and purchases.

**Key functionalities**

The website provides the following key functionalities:

* User authentication:
  + Users can create accounts with unique usernames and passwords.
  + A login system verifies user credentials using a password.
  + User sessions are managed to maintain login status.
* Item management:
  + The system stores details of available game items, including name, type, and price.
  + Items can be added to the database with their respective information.
* Purchase handling:
  + Users can view available items and their details.
  + A purchase mechanism allows users to buy items using their in-system balance.
  + The system updates user balances after successful purchases.
  + Purchase history is recorded, allowing users to track their transactions.

**Additional features**

* Search functionality: Users can search for specific items using a search bar.
* Purchase history: Users can view their purchase history, including item names, prices, and purchase dates.
* Balance display: Users can view their current balance on the website.

# V - Conclusion

Provide a brief overview of the main objectives and features of your game item trading system in this area. Stress how the design decisions you made (search, database system, user interface, etc.) helped you reach those objectives. You can also discuss any prospective advancements or upgrades in the future. As an illustration:  
  
"The goal of the Game Items Trading System (GITS) is to give players a convenient and safe way to exchange virtual goods. Using a MySQL database and web technologies like HTML, CSS, JavaScript, and maybe Node.js, GITS provides a user-friendly interface for item browsing, searching, and buying. The system's search capabilities, transaction history transparency, and responsive design all put the needs of the user first. In the future, developers may look into blockchain technology for improved security and ownership verification or integrate with certain game APIs."

# VI – Reference

<https://stackoverflow.com>

Github repo: https://github.com/COS30049-SUVHN/cos30049-fall2024-cos30049\_g2