

# Documentation: Player Equipment Management System

## Content

### 1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 References

### 2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

### 3. External Interface Requirements

- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

### 4. System Features

- 4.1 Inventory Management
- 4.2 Equipment Management
- 4.3 Other Management

### 5. Other Nonfunctional Requirements

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes
- 5.5 Business Rules

### 6. Other Requirements

Name	Date	Reason for changes	Version
Marcelina Górka and Michalina Górka	30.04.2024	None	1.01

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to provide a comprehensive overview of the Player Equipment Management System, including its functionality, features, and usage.

## 1.2 Document Conventions

- **Code Formatting:** Code snippets are formatted using monospace font.
- **User Input:** User input is represented within colon `:`.
- **System Output:** System output is presented in regular.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for developers, testers, and stakeholders involved in the development and deployment of the Player Equipment Management System. It is recommended to read through the entire document to gain a complete understanding of the system.

## 1.4 Product Scope

The Player Equipment Management System is designed to provide players with a user-friendly interface to manage their in-game inventory and equipment. It offers various functionalities such as buying and selling items, equipping gear, leveling up equipment, and interacting with a virtual shop.

## 1.5 References

- No external references are currently applicable.

# 2. Overall Description

## 2.1 Product Perspective

The Player Equipment Management System operates as a standalone application within the context of a larger gaming environment. It interacts with the game's core mechanics to provide players with inventory management capabilities.

## **2.2 Product Functions**

- Inventory management
- Buying, selling and deleting items
- Equipping gear
- Leveling up equipment
- Interacting with a virtual shop
- Displaying details, statistics and shop

## **2.3 User Classes and Characteristics**

Users of the system include players of the game who wish to manage their inventory and equipment efficiently. They may vary in experience level and familiarity with the game's mechanics.

## **2.4 Operating Environment**

The system is designed to run on platforms compatible with C++ programming language and standard input/output streams.

## **2.5 Design and Implementation Constraints**

- The system is implemented in C++.
- It may have dependencies on specific libraries or system functions for screen clearing and user input/output.

## **2.6 User Documentation**

User documentation may include in-game tutorials, help menus, and tooltips to guide players through the system's functionality.

## **2.7 Assumptions and Dependencies**

The system assumes a basic understanding of gaming concepts such as inventory management and equipment upgrading. Dependencies may include the availability of hardware resources and system libraries.

## **3. External Interface Requirements**

### **3.1 User Interfaces**

- Command-line interface for user interaction
- Menu-based navigation system

### **3.2 Hardware Interfaces**

No specific hardware interfaces are required.

### **3.3 Software Interfaces**

- C++ Standard Library for input/output operations
- System functions for screen clearing

### **3.4 Communications Interfaces**

No external communication interfaces are required.

## **4. System Features**

### **Show Equipment (U1)**

1. Objective - the customer is able to see his inventory
2. Priority - high
3. Source - user
4. Actors - customer
5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. System shows equipment after user choose option from menu
  - 5.2. Alternative Flow(s) - None
6. Includes
  - 6.1. Display
7. Preconditions - User run code
8. Post conditions - User can see items in their inventory
9. Notes/Issues - None

### **Display shop (U2)**

1. Objective - the customer see, what he can buy from shop

2. Priority - high
3. Source - user
4. Actors - customer
5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “o” from menu
    - 5.1.2. System show shop
  - 5.2. Alternative Flow(s) - None
6. Includes
  - 6.1. Display
7. Preconditions - User run code
8. Post conditions - User see all options from shop
9. Notes/Issues - None

### **Move (U3)**

1. Objective - the customer can move items
2. Priority - medium
3. Source - user
4. Actors - customer
5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “m” from menu
    - 5.1.2. Customer enters row and column of item that he want to move
    - 5.1.3. Customer enters row and column of place that he want an item to be moved
    - 5.1.4. System checks if there is item on given row and column
    - 5.1.5. System moves item
  - 5.2. Alternative Flow(s) - None
6. Includes
  - 6.1. Display
7. Preconditions - None
8. Post conditions - User has moved item
9. Notes/Issues - None

### **Buy (U4)**

1. Objective - the customer is buying item from shop
2. Priority - medium
3. Source - user
4. Actors - customer
5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “b” from menu
    - 5.1.2. Customer enters category of item that he want to buy
    - 5.1.3. Customer enters item he want to buy (it he choose weapon or armor he also enters rarity)

- 5.1.4. System decreases amount of user's gold by item's price
  - 5.1.5. System gives item to user
- 5.2. Alternative Flow(s)
  - 5.2.1. At point 5.1.2 if user enters category incorrectly
    - 5.2.1.1. Return to point 1.
  - 5.2.2. At point 5.1.3 if user enters item incorrectly
    - 5.2.2.1. An error message is displayed "Item not found in shop."
    - 5.2.2.2. Return to point 1.
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is enough money to buy item
- 8. Post conditions - Player have item he bought
- 9. Notes/Issues - None

### **Sell item (U5)**

- 1. Objective - the customer is selling item from his inventory
- 2. Priority - medium
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option "s" from menu
    - 5.1.2. Customer enters row and column of item that he want to sell
    - 5.1.3. System checks if there is item on given row and column
    - 5.1.4. System divides price of item by 3/4 and cuts everything that is after coma
    - 5.1.5. System issues money to the user
  - 5.2. Alternative Flow(s) - At point 5.1.3 if on given row and column there is no item
    - 5.2.1. An error message is displayed "Item not found in your equipment."
    - 5.2.2. Return to point 1.
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is item on given row and column
- 8. Post conditions - Money has been given to the user and their amount of gold has been updated
- 9. Notes/Issues - None

### **Delete item (U6)**

- 1. Objective - the customer is deleting item from his inventory
- 2. Priority - medium
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow

- 5.1.1. Customer chooses option “d” from menu
  - 5.1.2. Customer enters row and column of item that he want to delete
  - 5.1.3. System checks if there is item on given row and column
  - 5.1.4. System deletes item
- 5.2. Alternative Flow(s) - At point 5.1.3 if on given row and column there is no item
  - 5.2.1. An error message is displayed “Item not found in your equipment.”
  - 5.2.2. Return to point 1.
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is item on given row and column
- 8. Post conditions - None
- 9. Notes/Issues - None

### **Display player statistics (U7)**

- 1. Objective - the customer can see player statistics
- 2. Priority - medium
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “p” from menu
    - 5.1.2. System displays statistics of player
  - 5.2. Alternative Flow(s) - None
- 6. Includes
  - 6.1. Display
- 7. Preconditions - None
- 8. Post conditions - None
- 9. Notes/Issues - None

### **Set main armor (U8)**

- 1. Objective - the customer can equip armor
- 2. Priority - medium
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “a” from menu
    - 5.1.2. Customer enters what he want to equip (helmet, chets, pants, boots)
    - 5.1.3. Customer enters row and column of item that he want to equip
    - 5.1.4. System checks if there is item on given row and column
    - 5.1.5. System checks if item is armor
    - 5.1.6. System equips item
  - 5.2. Alternative Flow(s)

- 5.2.1. At point 5.1.2 if customer enters it incorrectly
    - 5.2.1.1. Return to point 1.
  - 5.2.2. At point 5.1.4 if on given row and column there is no item
    - 5.2.2.1. Return to point 1.
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is item on given row and column
- 8. Post conditions - Item is equipped
- 9. Notes/Issues - there is no information if customer enters it incorrectly and 5.1.4 if on given row and column there is no item

### **Set main weapon (U9)**

- 1. Objective - the customer can set weapon as main
- 2. Priority - medium
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “w” from menu
    - 5.1.2. Customer enters row and column of item that he want to set as main weapon
    - 5.1.3. System checks if there is item on given row and column
    - 5.1.4. System checks if item is weapon
    - 5.1.5. System sets item
  - 5.2. Alternative Flow(s) - At point 5.1.3 and 5.1.4 if there is no item on given row and column or item is not weapon
    - 5.2.1. An error message is displayed “This is not a weapon!”
    - 5.2.2. Return to point 1.
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is item on given row and column
- 8. Post conditions - Item is equipped
- 9. Notes/Issues - None

### **Sort equipment (U10)**

- 1. Objective - the customer can sort equipment
- 2. Priority - low
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “r” from menu
    - 5.1.2. System sorts players inventory



- 5.2. Alternative Flow(s) - None
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There are items in players inventory.
- 8. Post conditions - Inventory is sorted
- 9. Notes/Issues - None

### **Extend equipment (U11)**

- 1. Objective - the customer can extend equipment
- 2. Priority - low
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “q” from menu
    - 5.1.2. System extends players inventory
  - 5.2. Alternative Flow(s) - None
- 6. Includes
  - 6.1. Display
- 7. Preconditions - None
- 8. Post conditions - Inventory is extended by one row and one column
- 9. Notes/Issues - None

### **Show details (U12)**

- 1. Objective - the customer can see details of item
- 2. Priority - low
- 3. Source - user
- 4. Actors - customer
- 5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option “t” from menu
    - 5.1.2. Customer enters row and column of item
    - 5.1.3. System shows details of item
  - 5.2. Alternative Flow(s) - At point 5.1.2 if on given row and column there is no item
    - 5.2.1. An error message is displayed “Item not found at position (row, column) in the player’s equipment.”
    - 5.2.2. Return to point 1
- 6. Includes
  - 6.1. Display
- 7. Preconditions - There is item on given row and column
- 8. Post conditions - Player can see item’s details
- 9. Notes/Issues - None

### **Level up (U13)**

1. Objective - the customer can level up item
2. Priority - low
3. Source - user
4. Actors - customer
5. Flow Of Events
  - 5.1. Basic Flow
    - 5.1.1. Customer chooses option "l" from menu
    - 5.1.2. Customer enters row and column of item
    - 5.1.3. System decreases player's gold by 20
    - 5.1.4. System levels up item
  - 5.2. Alternative Flow(s) - None
6. Includes
  - 6.1. Display
7. Preconditions - None
8. Post conditions - Player can level up item by 20 gold
9. Notes/Issues - Player can enter food (that has no level) or empty slot and even then the system "levels up" and takes 20 gold

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

- System response time should be minimal for smooth user experience.
- Memory usage should be optimized to prevent excessive resource consumption.

### **5.2 Safety Requirements**

- The system should handle user input safely to prevent crashes or unexpected behavior.

### **5.3 Security Requirements**

- No specific security requirements are applicable.

### **5.4 Software Quality Attributes**

- Reliability: The system should function consistently without unexpected errors.
- Maintainability: The codebase should be well-structured and easily modifiable for future updates.
- Usability: The user interface should be intuitive and easy to navigate.

## 5.5 Business Rules

- Transactions should be accurately recorded and reflected in the player's inventory.
- Players should not be able to perform actions that would result in unfair advantages or exploits.

## 6. Other Requirements

No other specific requirements are currently identified.

Diagram UML

