

Titanic Treasures

Implementation Document, Version 1

April 24, 2018

Group Members:

Jacob McDonald

Tom Jackson

Clayton Richardson

Steven Romero

Lee Ward

Group Number: 7

Lab Instructor:

Dr. Ibrahim Abdoulahi

Task and Role Assignments

Requirements Phase:

Tom Jackson: User Interface
Jacob McDonald: User Interface/Specific Requirements
Clayton Richardson: Specific Requirements
Steven Romero: Specific Requirements
Lee Ward: Use Case Diagram and Specific Requirements

Design Phase:

Tom Jackson: Class Diagrams
Jacob McDonald: Sequence Diagrams
Clayton Richardson: Overview Class Diagram
Steven Romero: Class Diagrams
Lee Ward: Database and Class Design

Implementation Phase:

Tom Jackson: Graphic Design
Jacob McDonald: GUI Design
Clayton Richardson: PHP-HTML Interactions
Steven Romero: CSS, HTML, JS Design and Development, PHP Backend Intergration
Lee Ward: Database Design and Construction, Backend Design and Construction, PHP Backend Intergration

Pluto Group Account

username: law448

password: ab1234!

MySQL Group Account

username: dcsp07

password: iceberg

Web Accounts

admin username: spaceGato

 Password: abc

normal username: wrekker

 Password: 123

Database Design (differences from design)

(“*” = Primary Key, “#” = Foreign Key)

USER Table

 billingState

ORDERS Table

 shippingName

 orderTotal

ORDERS_ITEMS Table

 itemName

 itemTotal

 (Removed itemID from items table)

Changes were made to the database design to allow for data from one particular table to be more useful. If a user decided to delete an item from his/her inventory, a conflict would occur between the ITEMS and ORDERS_ITEMS tables. Our solution is to require the itemName and the collective price of those items into the table. Now, the ORDERS_ITEMS table is storing information that make it not rely on ITEMS table.

Source Code

Cart: The calculateCartTotal() function was not implemented. Instead, the cartTotal is calculated within the addItemToCart function.

Item: Due to the nature of the PHP language, static functions were implemented instead of overloaded constructors.

Order: An associative array was added to the Order class to add functionality to the class similar to the Cart class. This array is not meant to be changed beyond initialization due to the static

quality of an order. Static functions were too added to this class to take the place of our overloaded constructors.

Search: No deviations from the design were implemented.

User: No deviations from the design were implemented.

Requirements Not Meet

CheckOut

Add Item to Cart (Required)

Add Item to Inventory

Delete Item to Inventory

Mediums Not Meet

Browse Store

Recover Password

Add Avatar

Delete Avatar

Add Item Comments

Delete Item Comments

Rate Item

These requirements were not met due to time restraints and our ambitious design. Currently, the backend code is complete and tested. A majority of our webpages have been created, but not all PHP implementation of those sites have been completed.