THE PC PARTS PLACE

Dustin Johnson

CST-451 Capstone Project Final Plan Document

Grand Canyon University

Instructor: Professor Amr Elchoumi

Revision: 1.0

Date: 7/31/2022

**ABSTRACT**

The project is to cover creating the necessary features and abilities of an e-commerce site. This document will introduce the finalized design plan prior to implementation and testing takes place. This plan is designed to share with stakeholders and includes a change log template to document any required changes.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

## **Final Plan**

The PC Parts Place

Prepared by Dustin Johnson

|  |
| --- |
| Design Planning Summary |

1. This project is to design and implement an alternative solution to purchasing quality computer system components. The need for this is to give a service that is not complex and does not require extensive knowledge of compatibility of components in addition to more flexible customer service situations than other companies can provide. The problems that many customers have faced with returns and manufacturer repairs under warranty is a goal to be solved with this while still providing competitive pricing within the market.

|  |
| --- |
| Overview of Design Concepts |

1. The project is a C# web application following .NET MVC protocols. It will have a bootstrap framework for beautification comprising views for browsing all products, viewing individual product pages, creating a user account, logging into the site, a shopping cart, and a checkout process to create orders. Customer service queries are directed through a form that submits an email to the administrator to send out to a customer service rep.
2. This is a mockup of browsing all products

Shape, polygon

Description automatically generated

1. This image is a mockup of the layout of viewing an individual product. Graphical user interface, text, application

   Description automatically generated
2. This is a mockup of the layout of the login form

Graphical user interface, text, application

Description automatically generated

1. This is a mockup of the create an account formGraphical user interface

   Description automatically generated
2. This is a mockup of submitting a request for customer serviceGraphical user interface, text, application

   Description automatically generated
3. This is a mockup of what the cart might look likeGraphical user interface, application

   Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | Customers can register with site |  | Dustin | ToDo |  |
| 2 | Customers can add items to cart |  | Dustin | ToDo |  |
| 3 | Customers can checkout |  | Dustin | ToDo |  |
| 4 | Order information can be displayed |  | Dustin | ToDo |  |
| 5 | Permissions are working correctly |  | Dustin | ToDo |  |
| 6 | Customer Service requests can be submitted |  | Dustin | ToDo |  |

|  |
| --- |
| Detailed Solution Architecture |

The project will be broken down into multiple layers. There should be a solution for the Models and DataModels as well as the API used to make the calls to the database. The business logic should exist in the API solution. In the MVC solution, there should exist localized models built specifically for processing the DataModels into the views as needed for each view. The MVC level will contain manager, helper, and proxy files designed to make the calls to the API layer that handles the calls to retrieve data from the database.

Database Entity-Relationship Diagram



UML for skeleton of object models

Graphical user interface, table

Description automatically generated

Workflow structure throughout application

Diagram

Description automatically generated

Logical workflow diagrams

Add product to database

Text

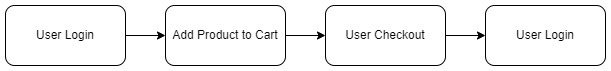
Description automatically generated

Admin Edit user accounts

Graphical user interface, text

Description automatically generated

Customer shopping flow



Remove product from database

Text, whiteboard

Description automatically generated

Customer register

Text, whiteboard

Description automatically generated

The approach to security will be taken in that each user will have their own level of access granted to them by the system administrator account. The roles will be the guiding force to what content can be seen, how the data is manipulated and what users have access to which things. There will not be a “guest checkout” feature, so all users must be logged in to add items to their cart or checkout. The architecture of the application will also provide security by separating the data layers and objects from the view objects and utilizing other n-tier layering techniques.

The physical system will look similar to this diagram. The user will use a device (most likely a laptop or phone) to connect to the internet and access the web application that is hosted through a web server that runs the application and communicates with a storage SQL server.Diagram

Description automatically generated

|  |
| --- |
| Hardware and Software Technologies |
| 1 – Visual Studio IDE |
| 2 – Microsoft SQL Server Manager |
| 3 – Draw.IO (Diagrams.net) |

|  |
| --- |
| Revision and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Editor** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |