Chapter 3 – Design

3.1 - Chapter Introduction

The design chapter discusses the proposed system's overall structure. This chapter describes the design processes, tools, and techniques used in the design phase, as well as the database of the system, as well as appropriate Use Case Diagrams, Entity Relationship Diagrams, and other UML Diagrams.

3.2 - Alternate Solution

Alternatives to the Web-Based Computer Hardware Purchasing and Troubleshooting Assistant management system include a mobile application, the use of a software collection, or the continued use of the current manual system.

3.2.1 - Stand-alone System

The stand-alone software is software that is not bundled with another piece of software and does not require an internet connection to run (work offline). It is not very useful for the Web-Based Computer Hardware Purchasing and Troubleshooting Assistant management system because outdoor service management cannot be carried out without the use of the internet, it is more expensive, and installation and maintenance are more difficult than with a Web-Based System.

3.2.2 - Collection of Software

The applications in the following list may be helpful in managing the Computer Hardware Purchasing and Troubleshooting Assistant management system.

- Appointment booking is a tool that assists in the scheduling of customer appointments.
- Inventory can manage individual objects as well as their corresponding categories, brands, and models.
- The warranty module facilitates the creation of warranty warnings. It displays the status of the goods' warranties as well as the remaining time on each item's warranty.

3.2.3 - Reason to Choose the Web-Based System

- The client's desire for a web-based system was unique.
- It's simple to keep track of the system's progress from any location.
- The system would be platform-independent.

- Because the database is centralized and everything is synchronized, maintenance is simple.
- Allow users to access the system at any time and from any location.
- It also allows for the usage of a wide range of devices to access the system.

3.3 - The Architectural Design of the System

As a Procedural Programming Design Concept, the system will be developed utilizing the Non-Object Oriented technique.

In a procedural language, a program is a list of statements, each of which instructs the computer to perform a certain task. It concentrates on the technique (function) and algorithm that are required to complete the derived computation. When a program grows in size, it is divided into functions, each with a distinct purpose. One of the fundamentals of structured programming is the division of the program into functions and modules.

Procedural oriented programming has several characteristics.

- The emphasis is on getting things done (Functions).
- The program is divided into many functions.
- Passing parameters across functions allows them to communicate with one another.
- Global variables are shared between functions.
- The procedure calls are based on the notion.
- When it comes to program design, it takes a top-down approach.

3.4 - Use Case Diagram for the Proposed System

The following is a high-level Use Case Diagram for the Proposed System.

The system's actors:

Admin: Create users and manage user permissions in the system.

Shop Manager: Manage all the orders made by the customers.

Inventory Manager: Manage the inventory of the store.

Delivery Manager: Manage all deliveries and register new courier companies

Technician: Respond to the customer's appointments and follow the job card.
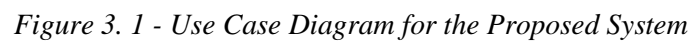
Customer: Make orders and make appointments.

Figure 3. 1 - Use Case Diagram for the Proposed System

3.5 - Use Case Narratives for the Proposed System

3.5.1 - Use Case Narratives for Login Module

Use-case Number	UC-04
Use-Case Name	Login
Priority	High
Actor	Admin, Shop Manager, Inventory Manager, Delivery Manager, Technician, Customer
Description	This use case describes how Shop Staff and Customers are logins to the System.
Precondition	All actors are properly registered to the system.
Post-condition	If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.
Basic course of Action	**User Action**	**System Response**
	1. The Actor is on the login page to log in to the system. 3. The Actor enters username and password, click on a Login Button.	2. The system promotes the Actor to enter the Username and Password. 4. The system verifies that all the filled have been filled out and valid. 5. The system successfully logged in the system. 6. Use case Exit
Alternate course of Action	4.1 If all fields are not filled out and not matched to the username and password the system notifies the actor a message "Invalid Username or Password" and then goes back or returns to step 3 of the basic course of Action to enter again.	

Table 3.1 - Use Case Narrative for Login Module

3.5.2 - Use Case Narratives for Insert Inventory Item

Use-case Number	UC-08
Use-Case Name	Insert Inventory Item
Priority	High
Actor	Inventory Manager
Description	This use case describes how to manage items in inventory
Precondition	None
Post-condition	If the use case was successful, the actor can add the product to the system and set it to sell to the customers.

Basic course of Action	**User Action**	**System Response**
	1. Actor clicks the "Add Item" in the Item Management section. 3. Actor fill the fields (Item Image, Item Name, Category, Brand, Model, SKU number, Insert variable specifications, reorder level, Unit price, and Sale price) 4. Actor click "Insert Item"	2. System prompts the form to insert details to the actor. 5. System validates the Item Name and SKU already in the system. 6. System display massage "New Item Insert"
Alternate course of Action	5.1. If an item is already inserted into the system display the popup message "This Item Already in the System".	

Table 3.2 - Use Case Narrative for Login Module

3.6 - ER Diagram for the Proposed System

The high-level ER Diagram below depicts the relationships between the proposed system's constituents. (Figure 3.2)
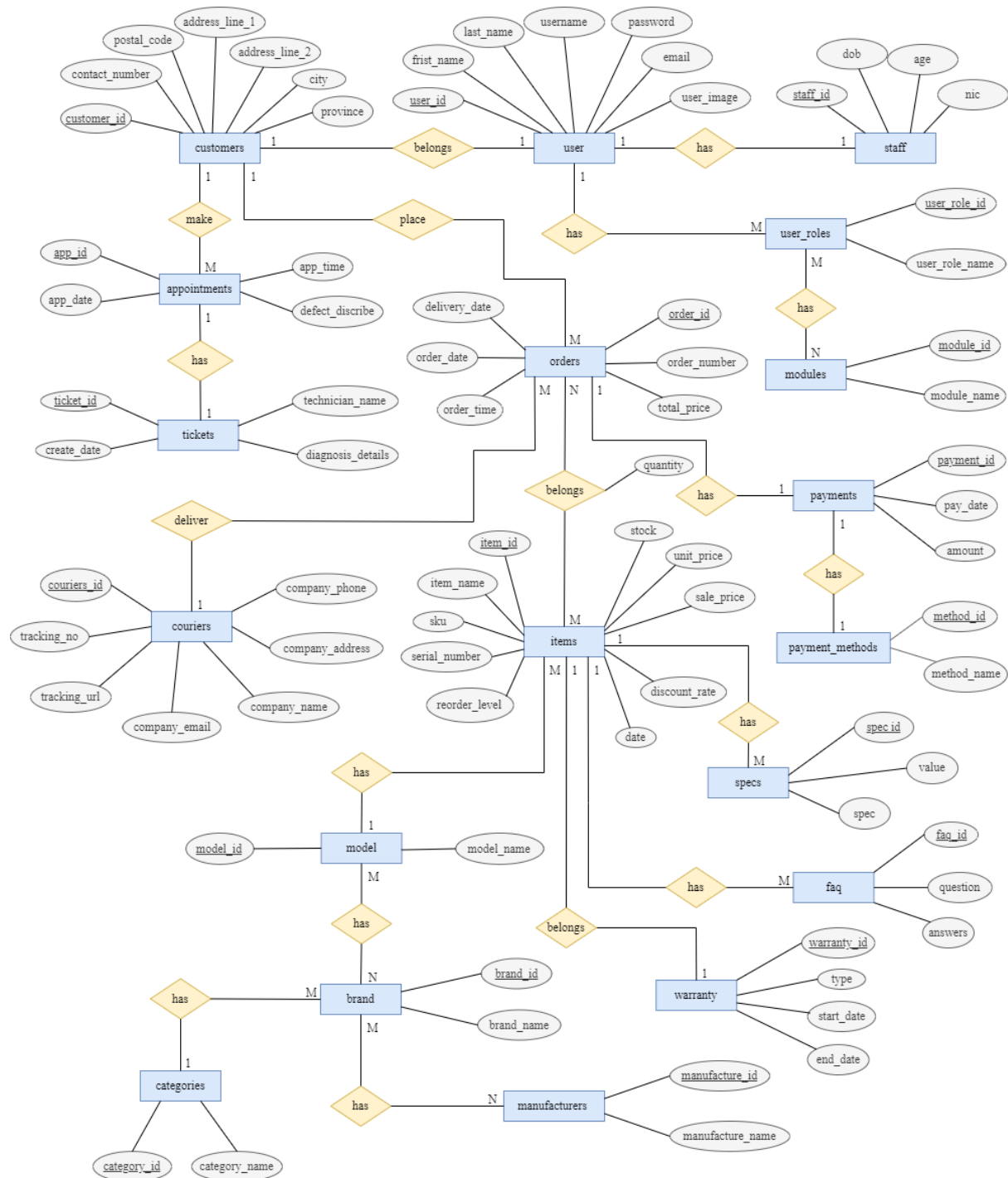
Figure 3. 2 – ER Diagram for Proposed System

3.7 - Activity Diagrams for the Proposed System

3.7.1 – Activity Diagram for Login

The activity diagram for creating workouts is shown in Figure 3.3.
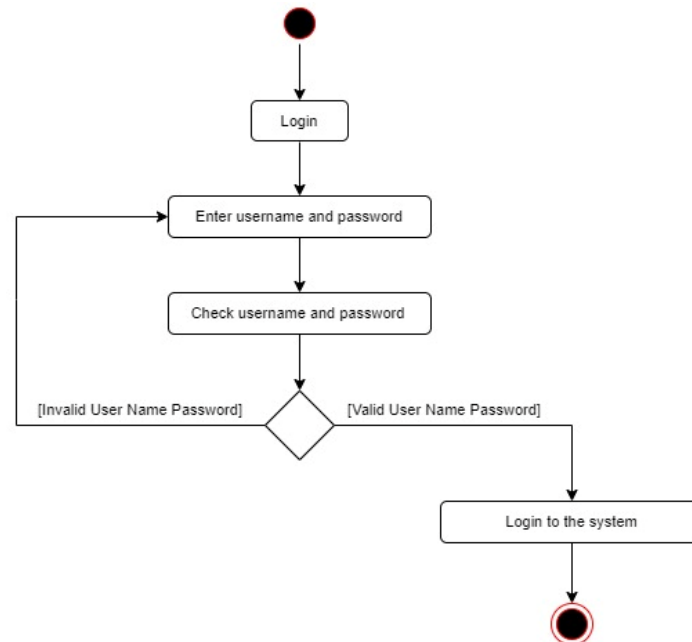
Figure 3. 3 – Activity Diagram for Login

3.7.1 – Activity Diagram for Add Item
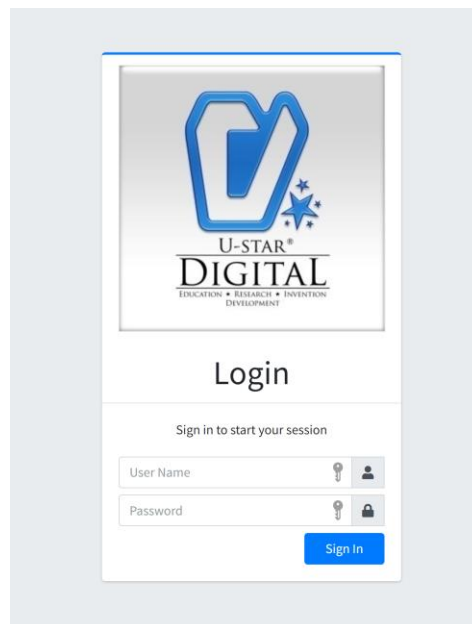
The activity diagram for add new item to inventory is shown in Figure 3.3.
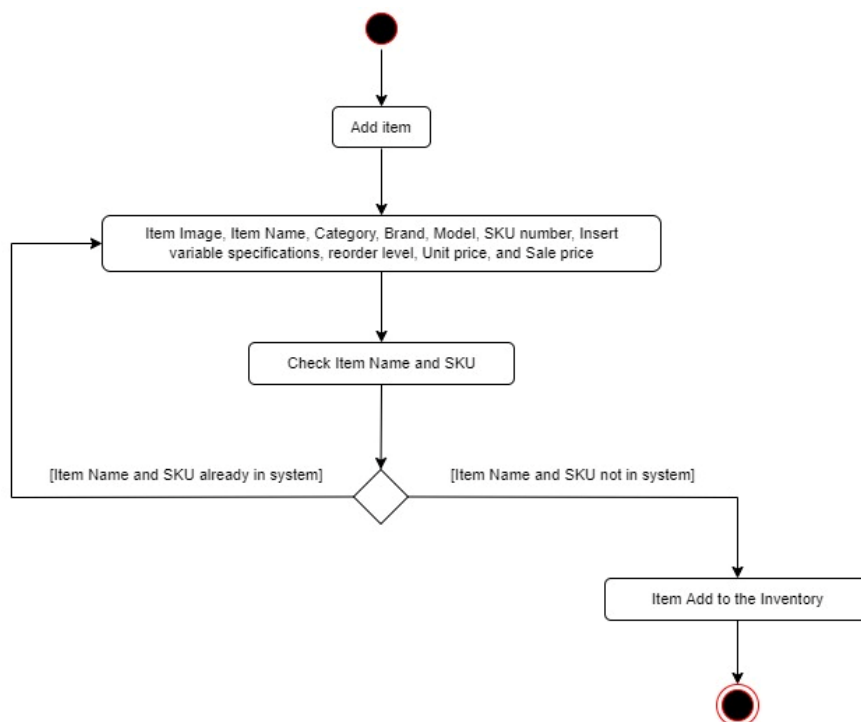
Figure 3. 3 – Activity Diagram for Add Item

3.9 – UI Designs for the Proposed System
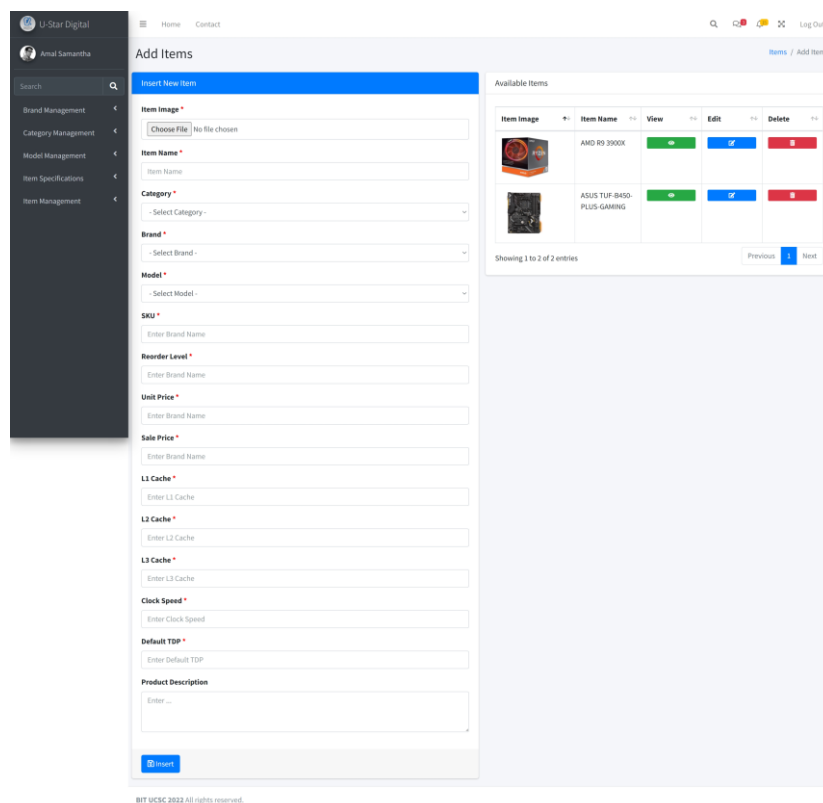
3.9.1 – Login Page UI Design

Figure 3. 5 – Login Form UI Design

3.9.2 – Add Item Page UI Design

Figure 3. 5 – Add Item UI Design

3.9.3 – Item Page View UI Design

Figure 3. 6 – Individual Item View UI Design