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

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 on 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:

Shop Manager:

Inventory Manager:

Delivery Manager:

Technician:

Customer:

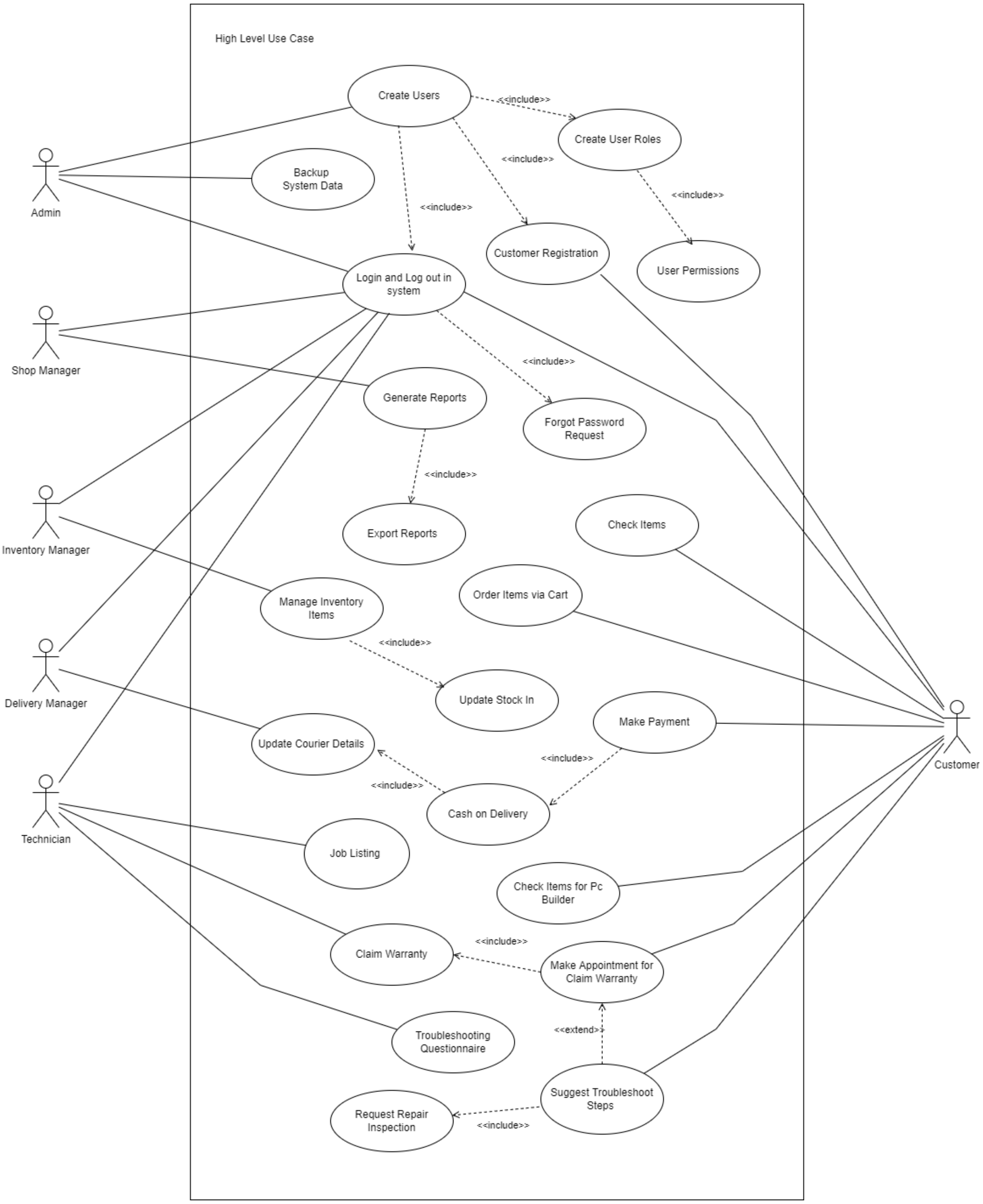

Figure 3. 1 - Use Case Diagram for the Proposed System

3.5 - Use Case Narratives for the Proposed System

3.5.1 - Use Case Narratives for Login Module

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)

3.7 - Activity Diagrams for the Proposed System

3.7.1 – Activity Diagram for Create Workouts

The activity diagram for creating workouts is shown in Figure 3.3.

3.8 – Database Diagrams for the Proposed System

3.9 – UI Designs for the Proposed System

3.9.1 - Login Page UI Design