**Final Project Statement of Work Template**

|  |
| --- |
| 7 |
| George Moshi |

**Group Number:**

**Group Leader:**

|  |
| --- |
| George Moshi (working on this project bymyself). |

**Names of Group Members:**

**Summary of what your program will do (be very specific): IT Management System**

The program is designed to manage an organization's IT assets efficiently. This system will allow IT administrators to track and manage hardware, software licenses, and user accounts, and locations. Acting as if it the program is similar to active directory. The system will leverage C++ programming features like class inheritance, composition, polymorphism, templates, STL containers, and dynamic memory management to create a comprehensive management tool. Alongside using searching and sorting tools and using standard template library alongside reading and writing data to file with using command line as the UI.

**Decomposition with Assignments:**

Use the space to list the decomposition or outline of your program. Next to each section, provide the name of the student who will be working on that portion.

|  |  |
| --- | --- |
| **Decomposition** | **Responsibility** |
| **\*\*Hardware Management\*\*:**  - \*\*Classes\*\*: Asset (base class), Computer, Printer, Router (derived classes using inheritance and polymorphism).  - \*\*Composition\*\*: Include a `Specifications` class used in each hardware class to hold details like processor speed, memory, storage, etc.  - \*\*Features\*\*: Register new hardware, update specifications, decommission assets.  - \*\*Search Function\*\*: Implement a function to search assets by type, model, or serial number.  \*\*Sort Function\*\*: Implement a function to sort software by expiration date or purchase date | George Moshi |
| **\*\*Software License Management\*\*:**  - \*\*Classes\*\*: Software (base class), OperatingSystem, BusinessApplication, UtilitySoftware (derived classes using inheritance).  - \*\*Features\*\*: Add new software licenses, update license details, check license cost, and remove expired licenses. | George Moshi |
| **. \*\*User Account Management\*\*:**  - \*\*Template Class\*\*: `User<T>` to manage different types of user roles like Administrator, StandardUser, Guest.  - \*\*Features\*\*: Add, update, or deactivate user accounts, reset passwords, and assign security groups | George Moshi |
| **\*\*Location Management \*\*:**  \*\*Classes and Structure:\*\* - \*\*Class\*\*: `Location`  - \*\*Attributes\*\*: `locationID`, `name`, `address`, `type` (e.g., physical, virtual, cloud-based)  - \*\*Methods\*\*: Add location, update location details, remove location  - \*\*Enhancements to Existing Classes\*\*:  - \*\*Hardware Class\*\*: Add an attribute for `locationID` which links each asset to a specific location.  - \*\*User Class\*\*: Optionally, track where users are primarily located to help manage software licensing or hardware allocation based on geographic or network location. | George Moshi |