**Documentation - Asset Tracking System**

**Non-Functional Requirements:**

1. The software system should be designed and developed to track assets withing the given company.
2. Information must be stored in a database which could be used by other company systems in the future.
3. All actions must be created to be used to manipulate all asset data. Actions included are add, view, edit or delete.
4. Software must be designed to automatically acknowledge key data regarding hardware from the system on which it is running. Key data should include (system name, model, manufacturer, type, IP Address, etc.)
5. Software can also include a component which can hold a capture of any physical stickers, which hold useful information.
6. Software must also include extra space for any data that the company may wish to provide regarding a specific asset.
7. Data should be intuitively corresponded.
8. Response time and data update must take no more than X milliseconds.
9. UI Design must be intuitive and easy to understand.
10. Manipulating data stored in the database must be very easy and fast to learn. \*User training time should be less than 30 minutes.
11. Operating should be on Microsoft Environment, which is able to run on a Windows 10 and 11 machine.
12. Software Must be able to request information via URL and return response converted from JSON to a class object
13. Software Must provide a feature for account creation, authentication, and authorization.

**Functional Requirements:**

**Introduction:** This section will cover all functionalities proposed from the client. Requirements will be fully detailed, which must be converted into a real implementation in the application. Validation will be held, to create test cases from specification, to assure implementation is according to specification.

**1. Asset Tracking – Used Components**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Status** | **Last Modified** |
| **1.1** For the creation of the database section the application team will use Microsoft SQL Server Management Studio to handle the Data. | **IMPLEMENTED** | **24/10/2022** |
| **1.2** The application itself will be developed via Visual Studio as an Application Project. | **IMPLEMENTED** | **24/10/2022** |
| **1.3** All functionalities must be technically described and for more complex use cases, UML diagrams must be used for visualization. | **IMPLEMENTED** | **24/10/2022** |

**2. Home Section – Layout and Functionality**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Status** | **Last Modified** |
| * 1. User must be able to see layout which consists of text and buttons. Text information must be relative to the function of each button visible on the canvas. | **IMPLEMENTED** | **24/10/2022** |
| **2.2** Each button must trigger an event from “***3. Event Triggering***”. | **IMPLEMENTED** | **24/10/2022** |
| **2.3** Empty non-usable buttons should not be visible. | **IMPLEMENTED** | **24/10/2022** |
| **2.4** All panels must contain a button directing to the “Home” page. | **IMPLEMENTED** | **30/10/2022** |
| **2.5** At start of application, algorithm must show details of machine on which it is running. If information could not be retrieved, it should output “Unknown:”. | **IMPLEMENTED** | **30/10/2022** |
| **2.5.1**  Information should be:  Name, Model, Manufacturer, IP Address | **IMPLEMENTED** | **30/10/2022** |

**3. Hardware Event Triggering – Functionality**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Status** | **Last Modified** |
| **JSON -> C# Class Object Converter** | **IMPLEMENTED** | **24/10/2022** |
| 3.1 **Add Asset Functionality**  *-This functionality shall offer the user to add a new asset and save it to the database.* | **IMPLEMENTED** | **24/10/2022** |
| 3.1.0 Inputs | **IMPLEMENTED** | **24/10/2022** |
| 3.1.1  Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **24/10/2022** |
| 3.1.2  String iname – Information from input field “name” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.3  String imodel – Information from input field “model’ | **IMPLEMENTED** | **24/10/2022** |
| 3.1.4  String imanufacturer – Information from input field “manufacturer” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.5  String itype – Information from input field “type” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.6  String iIPAddress – Information from input field “IP Address” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.7  String iPurchaseDate – Information from input field “Purchase Date” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.8  String iExtraInfo – Information from input field “Extra Information” | **IMPLEMENTED** | **24/10/2022** |
| 3.1.9  String oResponse – Text indicating  OPTION 1: “You have successfully added an asset!”  *OR*  OPTION 2: “Please Add All the Required Information”. | **IMPLEMENTED** | **24/10/2022** |
| 3.1.10 Button “Add” – Triggering the writing to the database. | **IMPLEMENTED** | **24/10/2022** |
| 3.1.11 Expected Behavior | **IMPLEMENTED** | **24/10/2022** |
| 3.1.12 User must be able to add an asset by clicking a button “Add a new asset”. This shall trigger an event, leading to a panel, which visualizes input boxes for the needed information and button “Add”. | **IMPLEMENTED** | **24/10/2022** |
| 3.1.13 After “Add Asset” panel is visualized, input boxes shall be visible for *system name, model, manufacturer, type, IP Address, purchase date* (optional), *extra information* (optional). | **IMPLEMENTED** | **24/10/2022** |
| 3.1.14 Button “Add” must trigger an event, which shall check if all “required” information is written. If it is successful, information shall be sent to the database and *oResponse* text shall indicate OPTION 1 (from 3.1.9). If information is not written, triggering shall be declined and *oResponse* text shall indicate OPTION 2. | **IMPLEMENTED** | **24/10/2022** |
| 3.1.15 Use Case Diagram  **The diagram above provides information and illustrates the process of an “Add” event triggering. The user shall input the required information and via clicking a button send a request for writing into the database. A response will be returned depending on the status of the event.** | **IMPLEMENTED** | **24/10/2022** |
| 3.2 **View Asset Functionality**  *-This functionality shall offer the user to view an asset/asset from the database.* | **IMPLEMENTED** | **25/11/2022** |
| 3.2.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 3.2.1 Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **25/11/2022** |
| 3.2.4 Results will be displayed in the ListView1 section. Providing information for all columns. | **IMPLEMENTED** | **25/11/2022** |
| 3.2.6 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 3.2.7 User must be able to view a specific asset or all which are stored in the database. A button must be used to check what the user has chosen. | **IMPLEMENTED** | **25/11/2022** |
| 3.2.10 Button “View” must be triggered to output the results from the search by requesting information from database. | **IMPLEMENTED** | **25/11/2022** |
| 3.2.11  **The diagram above provides information and illustrates the process of viewing an item. User must click view items and system shall return and expose all items from the database at runtime.** | **IMPLEMENTED** | **25/11/2022** |
| **3.3 Edit Asset Functionality**  *-This functionality shall offer the user to edit an asset and update it to the database. Edit should also provide the user the possibility to add a software component and link it to the HW item.* | **IMPLEMENTED** | **25/11/2022** |
| 3.3.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 3.1.1  Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **25/11/2022** |
| 3.3.2  Int iId – Information from input field “ID” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.2.1  Int softwareid – This will be used to link the ID from the software database, if hardware has software information connected |  |  |
| 3.3.3  String iname – Information from input field “name” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.4  String imodel – Information from input field “model’ | **IMPLEMENTED** | **25/11/2022** |
| 3.3.5  String imanufacturer – Information from input field “manufacturer” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.6  String itype – Information from input field “type” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.7  String iIPAddress – Information from input field “IP Address” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.8  String iPurchaseDate – Information from input field “Purchase Date” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.9  String iExtraInfo – Information from input field “Extra Information” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.10  String oResponse – Message Alert Box indicating status of event | **IMPLEMENTED** | **25/11/2022** |
| 3.3.11 Button “Update” – Triggering the writing to the database. | **IMPLEMENTED** | **25/11/2022** |
| 3.3.12 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 3.3.13 The user must fill all required input field, otherwise triggering event from Button “Update”, must not be possible. | **IMPLEMENTED** | **25/11/2022** |
| 3.3.14 Button “Update” must check for (3.3.13). If information is fulfilled, event shall trigger updating the asset with specific IId (3.3.2). | **IMPLEMENTED** | **25/11/2022** |
| 3.3.15 If after requesting to update from database, asset with “iId” could not be found, oResponse must output “Please select an item!” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.16 If after requesting to update from database, asset with “iId” is found and the update is successful, oResonse must output “Asset Successfully Updated” | **IMPLEMENTED** | **25/11/2022** |
| 3.3.17  The user can click “Add Software” to be directed to the software functionality (4.Software Event Triggering – Functionality), to add and automatically link software asset to current selected hardware asset. |  |  |
| 3.3.18  **The diagram above provides information and illustrates the process of updating an item. The user shall trigger a button “Update”, which will check if the user has selected an item. If the user has not selected an item a message box shall appear indicating that event could not be successful, and user must select an item. If the user has selected an item, the application will provide fields for data and also give an opportunity to add a software asset which shall automatically be linked to the given hardware asset.** | **IMPLEMENTED** | **25/11/2022** |
| **3.4 Delete Asset Functionality**  *-This functionality shall offer the user to delete an asset from the database.* | **IMPLEMENTED** | **25/11/2022** |
| 3.4.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 3.4.1 Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **25/11/2022** |
| 3.4.2 Int iId – The variable in which the user will enter an Id, to find a specific item. This id is automatically parsed, when the user selects an item from the list. | **IMPLEMENTED** | **25/11/2022** |
| 3.4.3 Button “Delete” – Will trigger a delete request in the database, looking for a specific asset with “iId” | **IMPLEMENTED** | **25/11/2022** |
| 3.4.5  String oResponse – Text indicating result of event | **IMPLEMENTED** | **25/11/2022** |
| 3.4.4 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 3.4.5 Button “Delete” shall trigger a request to delete item from the database ONLY if iId is found in the database. If deletion is successful oResponse must output “Successfully Deleted the Asset”. If iid is not found or null, request shall not be triggered and oResponse must output “Something went wrong.” Or “Please select an item!” | **IMPLEMENTED** | **25/11/2022** |
| 3.4.6  **The diagram above provides information and illustrates the process of deleting an item. The user must have selected an item from the list view in order to trigger a delete event.** | **FUTURE** | **FUTURE** |
| 4**. Software Event Triggering – Functionality** |  |  |
| 4.1 **Add Asset Functionality**  *-This functionality shall offer the user to add a new asset and save it to the database.* | **IMPLEMENTED** | **24/10/2022** |
| 4.1.0 Inputs | **IMPLEMENTED** | **24/10/2022** |
| 4.1.1  Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **24/10/2022** |
| 4.1.2  String inamesw – Information from input field “name” |  |  |
| 4.1.3  String iversionsw – Information from input field “OS Version | **IMPLEMENTED** | **24/10/2022** |
| 4.1.4  String imanusw – Information from input field “OS Manufacturer” | **IMPLEMENTED** | **24/10/2022** |
| 4.1.5  String hdwname– Information from input field “hardware name”.  This will be used to link information from Hardware Database (Optional) | **IMPLEMENTED** | **24/10/2022** |
| 4.1.9  String oResponse – Text indicating  OPTION 1: “You have successfully added an asset!”  *OR*  OPTION 2: “Please Add All the Required Information”.  \*oResponse is a message box. | **IMPLEMENTED** | **24/10/2022** |
| 4.1.10 Button “Add” – Triggering the writing to the database. | **IMPLEMENTED** | **24/10/2022** |
| 4.1.11 Expected Behavior | **IMPLEMENTED** | **24/10/2022** |
| 4.1.12 User must be able to add an asset by clicking a button “Add a new asset”. This shall trigger an event, leading to a panel, which visualizes input boxes for the needed information and button “Add”. | **IMPLEMENTED** | **24/10/2022** |
| 4.1.13 After “Add Asset” panel is visualized, input boxes shall be visible for *os name, os version, os manufacturer, hardware name (optional).* | **IMPLEMENTED** | **24/10/2022** |
| 4.1.14 Button “Add” must trigger an event, which shall check if all “required” information is written. If it is successful, information shall be sent to the database and *oResponse* text shall indicate OPTION 1 (from 4.1.9). If information is not written, triggering shall be declined and *oResponse* text shall indicate OPTION 2. | **IMPLEMENTED** | **24/10/2022** |
| 4.1.15    **The diagram above provides information and illustrates the process of an “Add” event triggering. The user shall input the required information and via clicking a button send a request for writing into the database. A response will be returned depending on the status of the event.** | **IMPLEMENTED** | **24/10/2022** |
| 4.2 **View Asset Functionality**  *-This functionality shall offer the user to view an asset/asset from the database.* | **IMPLEMENTED** | **24/10/2022** |
| 4.2.0 Inputs | **IMPLEMENTED** | **24/10/2022** |
| 4.2.1 Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **24/10/2022** |
| 4.2.4 Results will be displayed in the ListView2 section. Providing information for all columns. | **IMPLEMENTED** | **24/10/2022** |
| 4.2.6 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 4.2.7 User must be able to view a specific asset or all which are stored in the database. | **IMPLEMENTED** | **25/11/2022** |
| 4.2.10 Button “View” must be triggered to output the results from the search by requesting information from database. | **IMPLEMENTED** | **25/11/2022** |
| 4.2.11  **The diagram above provides information and illustrates the process of viewing an item. User must click view items and system shall return and expose all items from the database at runtime.** | **IMPLEMENTED** | **25/11/2022** |
| **4.3 Edit Asset Functionality**  *-This functionality shall offer the user to edit an asset and update it to the database.* | **IMPLEMENTED** | **25/11/2022** |
| 4.3.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 4.1.1  Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **25/11/2022** |
| 4..3.2  Int inamesw – Information from input field “OS name” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.3  String iversionsw – Information from input field “OS version” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.4  String imanusw – Information from input field “OS manufacturer” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.5  String hdwname – Information from input field “hardware” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.10  String oResponse – Message Alert Box indicating status of event | **IMPLEMENTED** | **25/11/2022** |
| 4.3.11 Button “Update” – Triggering the writing to the database. | **IMPLEMENTED** | **25/11/2022** |
| 4.3.12 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 4.3.13 The user must fill all required input fields, otherwise triggering event from Button “Update”, must not be possible. | **IMPLEMENTED** | **25/11/2022** |
| 4.3.14 Button “Update” must check for (4.3.13). If information is fulfilled, event shall trigger updating the asset with specific IId (4.3.2). | **IMPLEMENTED** | **25/11/2022** |
| 4.3.15 If after requesting to update from database, asset with “iId” could not be found, oResponse must output “Please select an item!” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.16 If after requesting to update from database, asset with “iId” is found and the update is successful, oResonse must output “Asset Successfully Updated” | **IMPLEMENTED** | **25/11/2022** |
| 4.3.17  **The diagram above provides information and illustrates the process of updating an item. The user shall trigger a button “Update”, which will check if the user has selected an item. If the user has not selected an item a message box shall appear indicating that event could not be successful, and user must select an item. If the user has selected an item, the application will provide fields for data and wait for the save trigger event.** | **IMPLEMENTED** | **25/11/2022** |
| **4.4 Delete Asset Functionality**  *-This functionality shall offer the user to delete an asset from the database.* | **IMPLEMENTED** | **25/11/2022** |
| 4.4.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 4.4.1 Prefix “I” shall be used for naming variables which are indicating inputs. Prefix “o” shall be used for naming variables which are indicating outputs. | **IMPLEMENTED** | **25/11/2022** |
| 4.4.2 Int iId – The variable in which the user will enter an Id, to find a specific item. This id is automatically parsed, when the user selects an item from the list. | **IMPLEMENTED** | **25/11/2022** |
| 4.4.3 Button “Delete” – Will trigger a delete request in the database, looking for a specific asset with “iId” | **IMPLEMENTED** | **25/11/2022** |
| 4.4.5  String oResponse – Text indicating result of event | **IMPLEMENTED** | **25/11/2022** |
| 4.4.4 Expected Behavior | **IMPLEMENTED** | **25/11/2022** |
| 4.4.5 Button “Delete” shall trigger a request to delete item from the database ONLY if iId is found in the database. If deletion is successful oResponse must output “Successfully Deleted the Asset”. If iid is not found or null, request shall not be triggered and oResponse must output “Something went wrong.” Or “Please select an item!” | **IMPLEMENTED** | **25/11/2022** |
| 4.4.6  **The diagram above provides information and illustrates the process of deleting an item. The user must have selected an item from the list view in order to trigger a delete event.** | **IMPLEMENTED** | **25/11/2022** |
| **5. User Account – Functionality** | **IMPLEMENTED** | **25/11/2022** |
| 5.1 – Overview  -In order to see and trigger events from (3,4), user must be logged into an account for security reasons. | **IMPLEMENTED** | **25/11/2022** |
| 5.1. User can log in into an existing account or create a new account. | **IMPLEMENTED** | **25/11/2022** |
| 5.2.0 Inputs | **IMPLEMENTED** | **25/11/2022** |
| 5.2.1.1 – Log In  Input field “Username” – Type text  Input field “Password” – Type password | **IMPLEMENTED** | **25/11/2022** |
| 5.2.1.2 – Register  Input field “Username” – Type text  Input field “Password” – Type password  Input field “Confirm Password” – Type password | **IMPLEMENTED** | **25/11/2022** |
| 5.2.1.3 – oResponse – Message box alerting status from even triggering. | **IMPLEMENTED** | **25/11/2022** |
| 5.2.2 Expected Behavior | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.1 – Register Functionality  User must click button “Register”, to be shown a panel for registering. Users must fill in all fields (username, password and confirm password) in order to successfully register. Password must be over or EQUAL 8 characters and must match with confirm password. | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.2 – Register Security  IF the user has successfully registered, the password must be encrypted/hashed and salted and then saved to the DB. Password must not be saved anywhere RAW. Only encrypted. | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.3 – Log in Functionality  The user must provide correct credentials in order to log in to account. If username or password is not correct oResponse (Message Box) must alert. The written password will be hashed, salted and verified and compared to the actual stored password against the username and id. If it is correct the user will successfully log in. | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.4 – Verification  After the user has logged in and is verification. All asset functionalities will be available for triggering. | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.5 – Change of Password  The user can click on the button “Account” which will direct him/her to a panel which gives the opportunity to change their password.  The user will need to input their current password and write a new password and confirm password. Authentication and security authorization will be executed as functionalities above (Hashing, comparing) | **IMPLEMENTED** | **02/12/2022** |
| 5.2.2.6 – Log out  The user can click on the button “Account” which will direct him/her to a panel which gives the opportunity to log out.  When the button is clicked all data related to the user account which was stored will be reset, and the user will be directed to the log in/register page. | **IMPLEMENTED** | **02/12/2022** |
| 5.2.  **The diagram above provides information and illustrates the process of creating an account.** | **IMPLEMENTED** | **02/12/2022** |
| 5.2.3    **The diagram above provides information and illustrates the process of authentication a log In event.** |  |  |
| **6. NVD Checking - Functionality** | **IMPLEMENTED** | **10/12/2022** |
| 6.1. Overview –  This section covers all functionality regarding the online request for NVD for any vulnerabilities regarding different operating system versions. | **IMPLEMENTED** | **10/12/2022** |
| 6.2.1  Buttons triggering NVD request for Windows:  8, 10, 11, selected database asset and current PC version. | **IMPLEMENTED** | **10/12/2022** |
| 6.2.2  Specific software asset information can be selected from list view, which can be used to trigger a request for NVD checking. This should only work with valid data and if data is not valid, message alert box should appear. | **IMPLEMENTED** | **10/12/2022** |
| 6.2.3  Data should be displayed in a list view, which includes Id, description and date published. | **IMPLEMENTED** | **10/12/2022** |
|  | **IMPLEMENTED** | **10/12/2022** |
| Data must be requested via URL request. Data will be returned in RAW JSON format and system must convert it to C# class object in order to use the data and output some of the collections to a list view | **IMPLEMENTED** | **11/12/2022** |