

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  | Milestone 1: Project Proposal, Prep (Planning Phase) | | |  |
|  |  | | |  |
|  | |  |  | |
|  | | John Keen johnkeenishere@gmail.com |  | |
|  | | December 22, 2023 —  Project Name:  Laptop Catalog —CST-451: Senior Project 1—Professor Brandon Bass —  Document Revision Number:  0 |  | |

# Abstract

This document outlines the details of a project where I will create a laptop catalog using ASP.NET. This would be useful for a company that manufactures laptops and wants customers to be able to search for laptops but would prefer that they buy them from an established retailer. I will create an API using C# so that retailers can access the inventory. I will also create an ASP.NET front end for CRUD operations to manage item in the laptop catalog.

|  |  |  |
| --- | --- | --- |
|  |  |  |

# Table Of Contents

[Milestone 1: Project Proposal, Prep (Planning Phase) i](#_Toc153975590)

[Abstract ii](#_Toc153975591)

[Table Of Contents iii](#_Toc153975592)

[Project Overview and Objectives 1](#_Toc153975593)

[Project Measure and Scope 2](#_Toc153975594)

[Project Completion 3](#_Toc153975595)

[Project Controls 4](#_Toc153975596)

[Project Schedule 4](#_Toc153975597)

[High Level Solution Design 5](#_Toc153975598)

[Issue Log: 6](#_Toc153975599)

[References 7](#_Toc153975600)

# Project Overview and Objectives

##### Problem and Background Stated

The purpose of this project will be to create a functioning catalog that shoppers can use to search for the latest laptops by a manufacturer. I will create a REST API using C# and ASP.NET in order to create a laptop item management catalog. A minimum viable product will be able to create, read, update and delete laptop inventory items. It will also allow users to search for laptops. If successful, I will have demonstrated the power of ASP.NET and C# REST API’s. Upon the completion of this project, I should be able to express my comprehension of DAO’s, interfaces, REST API’s and the MVC architecture. This project will fill my need of expressing an ability to perform in an entry level development position and the need of perspective employers to verify a potential employee’s skillset.

##### Christian Worldview

The Christian worldview emphasizes honesty and integrity. I will not take full credit for my senior project because it will be based upon technologies, activities and ideas I have encountered throughout my academic career. I will adhere to the ethical value of not using profanity or violent images in the creation of this web application. I will do my best to ensure a pleasant user experience.

##### Project Objectives

1. REST API can Create, Read, Update and Delete an inventory item
2. Manager can use UI to Create, Read, Update and Delete items
3. Users can search for laptop items.
4. A Manager should be able to login to edit items.

##### Challenges

I can hardly remember the tutorials that taught me how to create ASP.NET pages. All of the things I have mentioned I have learned and only done precisely once. This will increase the level of difficulty of creating the web application. My first challenge will be relearning how to create a user registration and login in ASP.NET. Then I will have to relearn how to create a REST API in C#. Lastly, I will create a front end that is capable of performing CRUD operations for inventory items. I am hoping that once I start learning how to accomplish all of this again that I effectively my skills to work and my memory regarding this topic comes back.

##### Benefits and Opportunities

The greatest benefit resulting from this project will be my ability to convey that I am capable of understanding and creating concepts such as REST APIS and MVC based web applications.

# Project Measure and Scope

The product will be a laptop catalog. The user will be able to register and login. Login will be secured using encryption and salting. Data will be passed to a MSSQL database as parameters. Security will be a concern because attackers may be able to take over the server that hosts the application if the registration and login inputs are not secured. Inputs used for CRUD operations must also be parameterized and secured in order to prevent malicious attacks by someone that has gained access to this functionality. The user’s vital information must be secure. My measure of success will be based on two sets of tests. The first set of tests will verify that data is being correctly stored and accessed in a database using a REST API. These tests will be conducted with Postman. The other tests for measuring success will be manual end user tests where functionality such as creating, reading, updating, deleting and searching are tested for integrity. Part of the end user testing include registering a user and logging in. I will check the data in the database to ensure that data is being inserted correctly into their respective fields.

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| John Keen | Product Owner,  Developer, End User | Write user stories, develop and test. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Create REST API | ASP.NET | Incomplete | 0 hours | $0 | Jan. 22, 2024 | Jan. 26, 2024 | 40 hours |  | Activities 1, 2, 3,4,5 and 7 of C# III |
| 2 | Create UI | ASP.NET | Incomplete | 0  Hours | $0 | Jan. 29,  2024 | Mar. 1,  2024 | 80 hours |  | Youtube, Activities 1, 2, 3,4,5 and 7 of C# III |

# Project Completion

The minimum viable product will be a laptop catalog that displays laptop inventory. The project is complete when it passes API testing and manual end user acceptance tests. There will be a standard of quality and performance where the application is free of glitches and bugs. This project should be monetarily free to create and will utilize open-source technologies. The project will not be hosted in the cloud as it could potentially be for demonstration purposes but will instead sit on my private server where it will not occur any charges. The completed project will be hosted on Github for others to view and download. An assumption that was made is that perspective employers would see an ASP.NET web application with crud functionality and a REST API as proof of skills that are relevant to a potential career opportunity. Documentation for the application must be complete in order to best demonstrate how this project conveys an aptitude in software development. The constraint of time must be met. This project needs to be completed prior to the beginning of my next class approximately two months after the end of this class.

|  |
| --- |
| Project Completion Criteria |
| 1 – Hosted on Github |
| 2 – API Creates Laptop |
| 3 – API Reads Laptop(s) |
| 4 – API Updates Laptop |
| 5 – API Deletes Laptop |
| 6 – User can register |
| 7 – User can login |
| 8 – Users can search |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | Potential lack of applicable knowledge. |  | Constraint |  | 12/23/23 |
| 2 | Relevant tutorials are readily available |  | Assumption |  | 12/23/23 |
| 3 | I will have enough time to complete the front end |  | Assumption |  | 12/23/23 |
| 4 | I have about two months to create a working Laptop catalog using ASP.NET. |  | Constraint |  | 12/23/23 |

# Project Controls

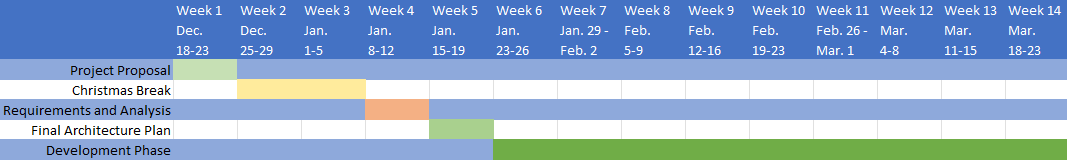
To streamline the process of development I will act as both a developer and an end user. There exists the possibility that I have a shortage of skills to complete the project. There are plenty of tutorials that can help address this potential problem. If my classes are rescheduled for some reason then I could potentially have the problem of running into time constraints. I currently have ample time determine how I will engineer my front end. I will create a schedule to ensure that I am on task and take breaks. Frequent breaks will help address the feeling of being stuck and over analyzing the task at hand.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| Adding too many features will increase complexity beyond completability. | Medium | My game may only be a partial success. | Clearly define requirements of features and stick to those features. Break the work up into a series of successes. | Seek simpler solutions on the internet. |
| I will spend too much time trying to complete one feature and not move on when needed. | Medium | Stress levels will increase and productivity will decrease | Create a Schedule. Take breaks. | Switch to another feature |
| I may not have the required skills to complete the project | Medium | Being stuck, frustrated and spending too much time on one feature. | Set time aside for learning. Seek relevant tutorials on the internet. | Talk to the professor about changing the project or taking the class over again. |
| The user interface navigation may need to be relearned. | Medium | Navigation may become frustrating | Consider multiple options and choose the least objectionable. | Find a project with similar navigation requirements and consider using the same technologies. |
| Accurately documenting code may be a challenge. | Medium | Code could become troublesome to troubleshoot. Documentation would not assist in troubleshooting code. It would also not accurately convey what was accomplished. | Update documentation to accurately reflect code decision changes. | Seek peer code review. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
| Self | One | Product Owner | Create the user stories that can be turned into wireframes. |
| Self | One | Developer | Develop each feature of the application. |
| Self | One | End User | Manually test each feature several times after it has been created prior to moving to the next phase of development. |

# Project Cost and Schedule

The application should be free to create and will utilize open-source technologies. The application should be completed by the end of the first week of CST-452. I will have approximately two months off between this class and the next class I have scheduled. The class CST-452 starts during the last week of my application security foundations class. This means that coding should be completed prior to starting my application security foundations class. I will attempt to complete the REST API for creating and verifying users and the part of the API for creating, reading, updating and deleting products finished by the end of Week 6. I will attempt to create the front end during my months off so that development in Senior Project II continues as planned. 

# High Level Solution Design

##### Introduction

The challenge is to provide a way to manage inventory while also providing a way for retailers to access inventory information. The API will allow retailers to access the laptop inventory information without having to access the part of the website vital to crud operations. This high-level solution design provides a separation of concerns. A separation of concern makes units easier to test and maintain while also simplifying code so that bugs can be easily isolated (Wikimedia Foundation, 2023). Information that will be stored will pertain to the manager of crud operations or the inventory. Information pertaining to the manager will include a username, first name, last name, email, phone, city, state, zip, street address and hashed password. Information pertaining to the inventory will contain information pertaining to the product id, photo, product name, product description, price, processor, ram, drive size, screen size, graphics card, weight and operating system.

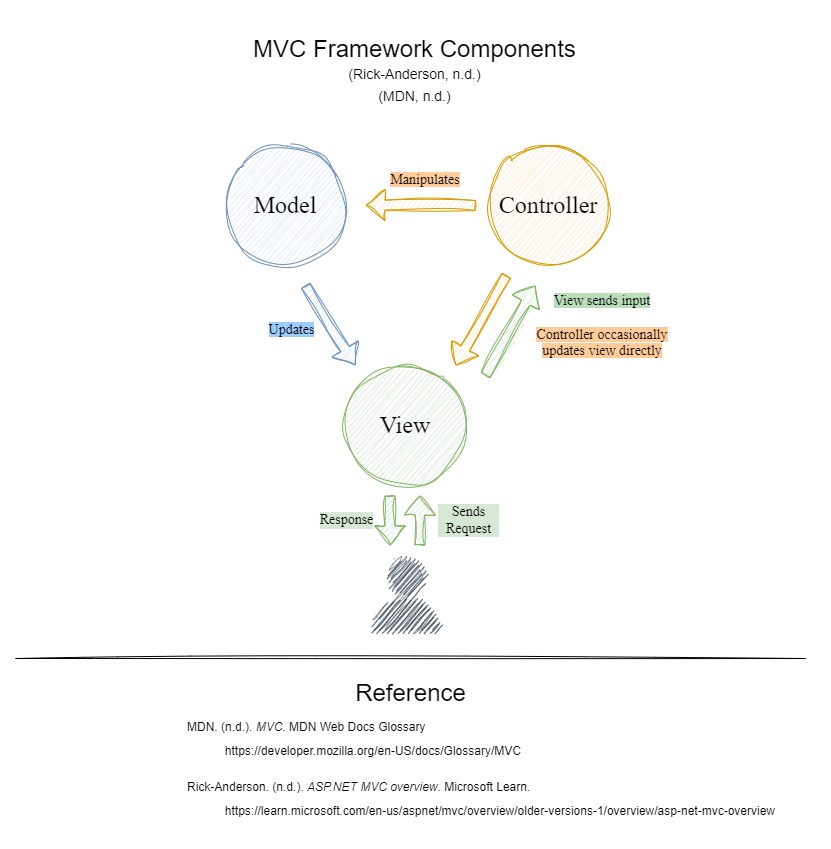


Figure 1 This diagram was created from another diagram that I used to illustrate Model View Controllers.

It was originally used in a discussion in C# III at GCU.

##### Solution

The solution will be to provide an API for the retailer and an MVC front end for managers. The MVC web application will have a presentation layer, data access layer, model and a database. There will not be very much business logic so the laptop DAO will be used in the LaptopController instead of providing a separate service layer. A service layer usually keeps business logic separate from the code in the controller so that it clearly and concisely handles views and models. For security reasons, the REST API will not allow users to create, update or delete inventory. This allows the API to be easily accessed without need for authentication. The API can be consumed without concern as weather or not the data on the server will be affected.

**API Methods:**

* Method: Index()
* URL: https://localhost:7269/
* Purpose: Return all items
* Method: SearchResults()
* URL: https://localhost:7269/searchresults/{searchTerm}
* Purpose: Return results for search where search term matches the product name
* Method: ShowOneLaptop
* URL: https://localhost:7269/showonelaptop/{id}
* Purpose: Return one item that matches the id

**Presentation Layer (ASP.NET MVC):**

Controllers:

LaptopController: Handles user input and manages communication between the View and the Model.

Responsible for receiving user requests from the View.

Views:

* Search.cshtml: Displays a list of laptops.
* Crud: Displays a list of laptops with buttons for crud functionality.
* Details.cshtml: Shows detailed information about a specific laptop.

**Data Access Layer:**

LaptopDAO (Data Access Object):

Responsible for interacting with the database to perform CRUD operations related to laptops.

Abstracts the database-specific logic from the rest of the application.

**Model:**

Laptop:

Represents the data structure for a laptop.

Includes properties such as product id, photo, product name, product description, price, processor, ram, drive size, screen size, graphics card, weight and operating system.

**Database:**

It will use a relational database (e.g., MSSQL) to store laptop information.

I will create a "Laptops" table with the following columns: product id, product name, product description, price, processor, ram, drive size, screen size, graphics card, weight and operating system.

**Dependency Injection:**

I will use Dependency Injection to inject the LaptopDAO into the Controllers.

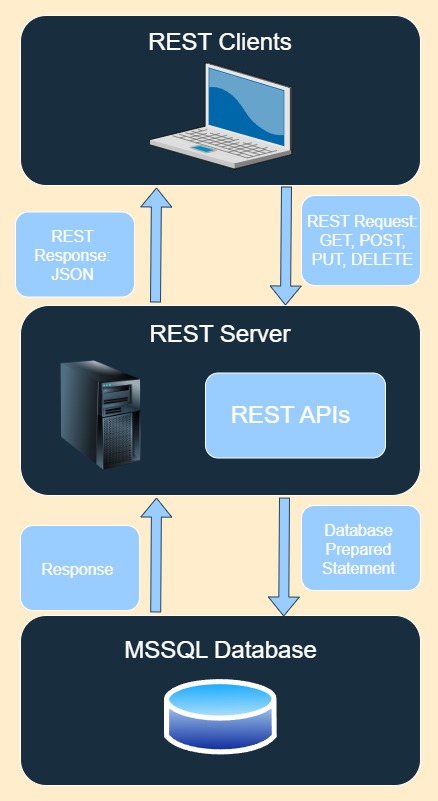


Figure 2 I created this image to demonstrate how REST APIs work.

In actuality the API will only GET and the database will be an MSSQL database.

# Issue Log

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | What is the issue? | How will this impact scope, schedule & cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

# References

Wikimedia Foundation. (2023, December 22). *Separation of concerns*. Wikipedia. https://en.wikipedia.org/wiki/Separation\_of\_concerns#:~:text=Separation%20of%20concerns%20results%20in,simplification%20and%20maintenance%20of%20code.