Pc Part App

- Database

- GUI

- C# code – middle layer

User Stories

- As a user I want to be able to see information on a specific part chosen (1)

- As a user I want to get a list of parts and order them by a specific attribute

- As a user I want to be able to put parts together to build my own machine

- As a user I want to be able to check for compatibility issues

- As a user I want to be able to login

- As a user in want to be able to save build for future references

- As a user I want to search for parts.

Product Backlog

An application that can retrieve parts from a database and able to get more information on them

Create Database

Link the database from SqlServer to Visual Studio2019 and C#

Understand Entity Framework Core better

Created Gui that interacts with database and code to provide a list of items that can be clicked for more information

Database

Table for PartsInformation (1)

Table for storing user login (3)

- username

- password / pin

Tables

- CPU (1)

- GPU

- Motherboard

- Memory/RAM

- Storage

- Power Supply

- Case

- Video Card

GUI

Window for Login (3)

Window for Home (2)

Window for parts retrieval (1)

Window for building machine

() <-- indicated order of importance regarding MVP 🡨- Definition of done

**Sprint 0**

**Sprint Goal**

Create database with one table and link it to C# and GUI to get an min viable product

Plan Steps for product and how I can break down project

Steps

- Create a Database from CPU

- Link it to visual Studio

- Link to GUI layer

- Create GUI to be able to retrieve database.

**Sprint Review**

-Created Database with a table named CPU and added Values in there

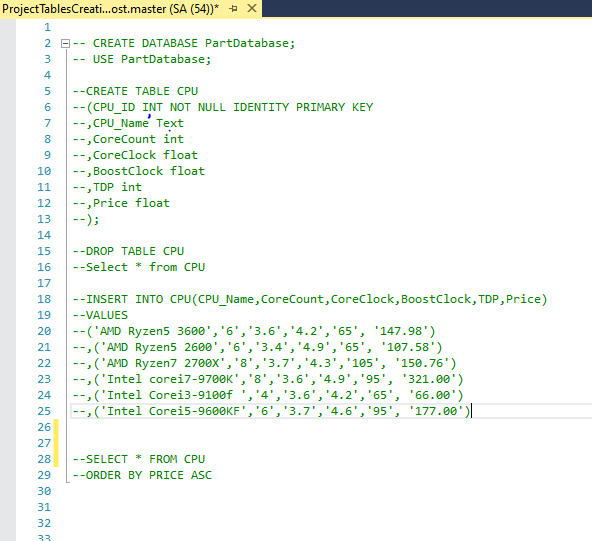
-Able to query the database

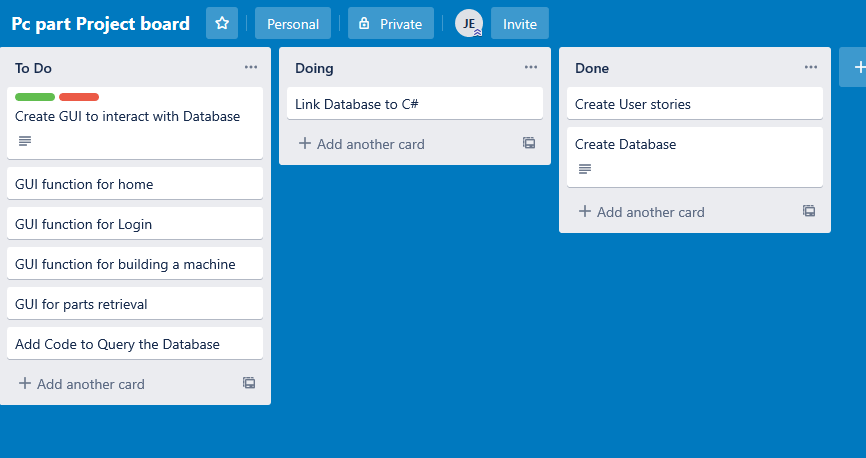
- Added Nish and Cathy as Collab’s on project on GitHub

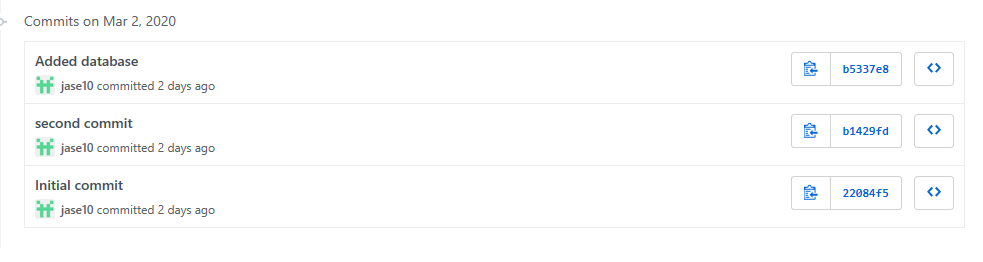
- Connected Database to Visual Studio

**Sprint Retrospective**

Started off with an abundance of ideas with a large project scope. Due to time constraints I narrowed it down so that I could focus on getting an MVP. What I did well was that I understood where I wanted to go with the project. I broke it down into smaller more manageable tasks so I can see progress after each sprint. I tend to overcomplicate the project.







**Sprint 1 Goal**

-As a user I want to be able to see information on a specific part chosen

-As a user I want to get a list of parts and order them by a specific attribute

- Create a Dataset so that a data can be retrieved in WPF

**Sprint 1 review**

-Created Query to filter the database

- Connected Windows to one Another

- Created Retrieval of parts page

-Designed the Gui

- Querying the database was a blocker to begin with but overcame it by creating window functionality

**Next Tasks**

- Next task is to finish adding a click method to list box so when an item is clicked its opened for more information

-This is broken down into

-Creating a table that can show the information when clicked

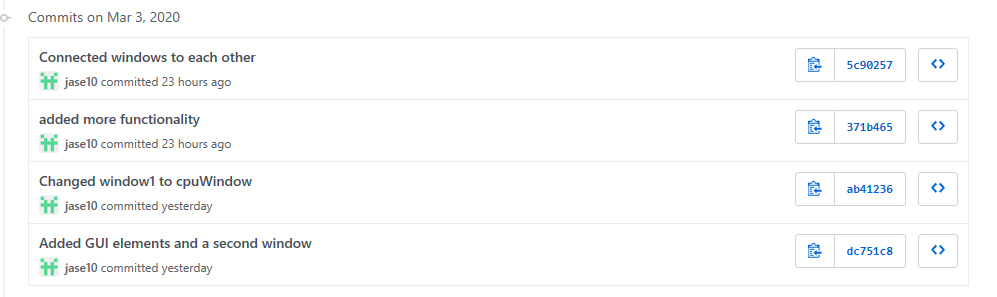
-Adding a click method so that user can interact with the items in the List

**Sprint Retrospective**

-Overall the mood for this sprint was not good as I could not query the database, so I had trouble getting my database layer into the project.

I decided to add window design and functionality instead.

What I did well was that I didn’t let a blocker affect the project as I left it and did other tasks instead.



**Sprint 2**

**Sprint 2 goal**

Connect Database

Add queries so I can get all items from a table.

Add click method so that the user can click to get more information.

