# **1** **Release Information**

|  |  |
| --- | --- |
| *Release* | *1* |
| *Leader* | *Edward Pazdzoria* |

Most of this iteration was focused on providing us a base to work with. The program can now read in .json files and has the beginnings of a GUI. We also have methods for returning the price of parts on Google Shopping.

# **2** **Progress**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stories and spike solutions worked on during this release** | | | | |
| **Issue #** | **Labels** | **Description** | **% Impl.** | **Notes, plan for completion** |
| #4 | *Story* | *Select guitar components in the GUI* | *30* | *Started, not all functionality present yet* |
| *#7* | *Story* | *GUI Implementation* | *60* | *GUI has nearly all components but lacks functionality* |
| *#2* | *Story* | *Assemble a collection of guitar components and images* | *60* | *All parts have at least one example, more will be added as necessary* |
| *#3* | *Story* | *Assembled image of all parts together* | *30* | *Got example assembly put together, need to attach it to the buttons for functionality* |
| *#11* | *Task* | *Image Creation* | *80* | *Most images are complete, may need to alter or add more later* |
| *#23* | *Task* | *Access part prices from JSON*  *files based on what parts are*  *selected* | *75* | *Part prices present in json file, just need to access the data when needed* |
| *#22* | *Task* | *Implement part selection in the GUI* | *65* | *Created buttons with images, still need functionality* |
| *#19* | *Task* | *Compile List of parts* | *60* | *All the parts are documented and for implemented to the basic function in the program* |
| *#12* | *Task* | *Implemented method for gathering the current price of a product* | *70* | *Method gathers price, but does not yet cache the price. We will want price caching in our final project, since we only have 1000 API requests total.* |
| *#14* | *Task* | *Modular image of guitar* | *30* | *Example assembly* |

|  |  |  |
| --- | --- | --- |
| **Accomplishments by engineer** | | |
| **Team Member** | **Total Time (hrs)** | **Stories/tasks implemented** |
| *Connor* | *14h* | Issues #20, #19, #14, #7, #2 |
| *EJ* | 13hrs | Issue #11, #22, #14, #4, #7 |
| *Josh* | 12hrs | Task #12, Story #15 |
| *Mukesh* | *11 hrs* | *Task #14 and #21 ,Story #4 and #7 implemented.* |
| TEAM TOTAL | ***50 hrs*** |  |

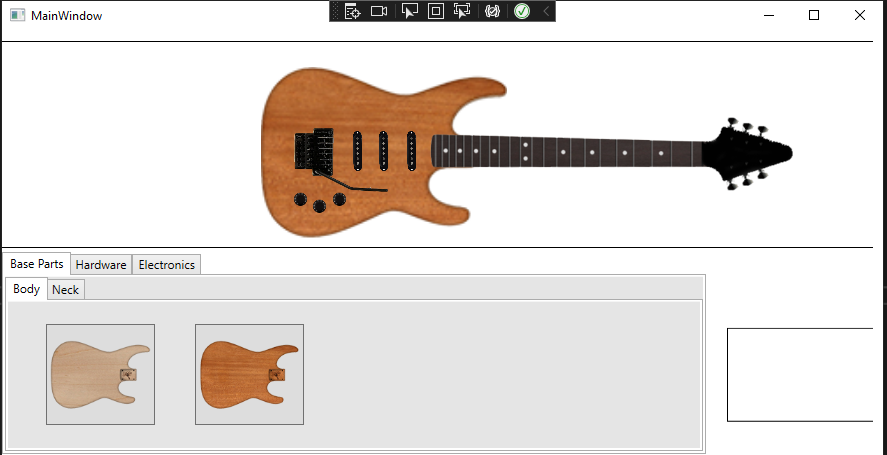
|  |  |
| --- | --- |
| **Document updates** | |
| **Document** | **Changes** |
| *starting\_parts\_list\_with\_links.docx* | *Document with parts, prices, and sources* |

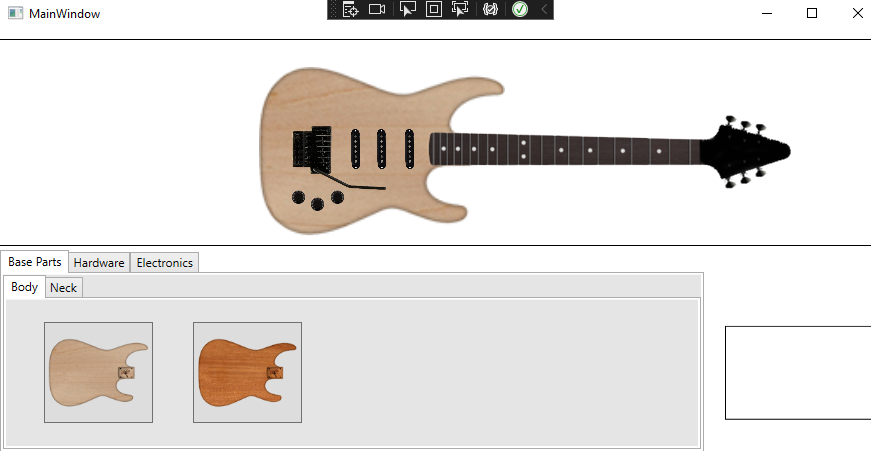
**3** **Issues**

Document anything that is not decided yet and needs to be resolved at some point.

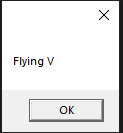
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue Number** | **Discovery**  **Date** | **Resolution**  **Date (Est.–Act.)** | **Responsible Person** | **Description (Prob / Resolution)** |
| 1 | 3/4 | 3/8 | Josh | Problem: I discovered there was no free API available that provides enough requests for us to use. I then attempted to web scrape the price from amazon, which would not have a limit, but found that Amazon’s website is designed to be very difficult to scrape.  Resolution: We will use the best API available to use (Price API), which gives us 1000 total price checks. This won’t be enough for us to update prices every time we run the program, we will implement a system that caches the part prices and only updates them on button press. |
| 2 | 3/5 | (3/6-3/6) | EJ | Problem: Realized that my initial set of images were out of scale and not oriented properly.  Resolution: place all images into one photoshop file and scale/rotate them with each other, then export each layer as its own png |
| 3 | 3/2 | (3/4-3/6) | Connor | Problem: JSON.Net would not function as described to deserialize data from the json file.  Resolution: Workaround was found, will make working with the json a little harder but still doable. |
| 4 | 3/5 | 3/6 | Mukesh | Problem: .Net Core 3.1 not working where i cant make GUI of the guitar main window  Resolution: Install .Net core 6.0 x64 bit and i can starting working in the GUI |

# **4** **Screen Shots**





The above images shows the progress we have made on GUI implementation.



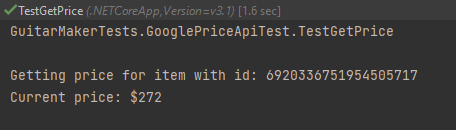
Information being read directly from json and displayed in a message box.

*Insert screen shots of functionality added for this release along with screen shots from spike solutions. Add short descriptions of all screen shots along with a short list of features illustrated by each. The goal is to convince your instructor that you completed the work documented above.*

Screenshot of Pricing API demo:

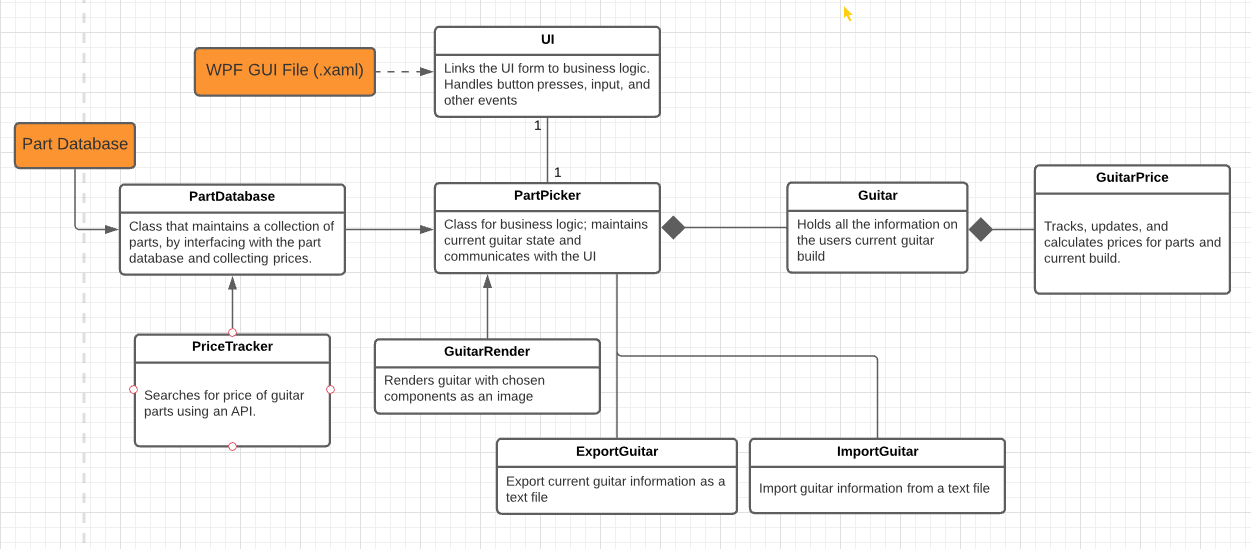


Screenshot showcasing images assembled in a document. This is an example of what the finished product should look like in our program.



This screenshot shows a demo of the pricing API. This output shows the program gathering the price of the item with the id: 6920336751954505717. Every item on the API has its own ID. In this case, this is the ID for a [Floyd Rose Original Limited 1984 Tremolo System](https://www.google.com/shopping/product/6920336751954505717?uule=w+CAIQICIfTmV3IFlvcmssTmV3IFlvcmssVW5pdGVkIFN0YXRlcw==&hl=en&gl=us).

# **5** **Updated Design**

**