**Product Backlog**

**for**

**Auto-Easy-Key**

**Version: 1.0**

**Date created: 10/3/21**

**Prepared by:**

**Hailey Vadnais**

**Stephen Behnke**

**Matthew Kotras**

**Sebastian Vang**

**Product Description**

Our product is a software that makes it easy to make scripts in AutoHotKey, a scripting language. Our target audience is anyone who has work that requires doing repetitive tasks on computers, but who are not great with computers. The problem with AutoHotKey is that inexperienced people are intimidated by coding and so they never get to experience the power of AutoHotKey in their daily lives. Our product solves this problem by turning scripting into a task that’s as easy to do as clicking a button.

**Functional Requirements**

Write up the functional requirements for your project using multiple user stories, each of which follows the template:

As a [kind of stakeholder],  
So that [I can achieve some goal],   
I want to [do some task]

Each of your user stories must also:

1. Be SMART. Pay special attention to *Specific* and *Measurable*. If your user stories have any ambiguity about this, further clarify them until they are clearly specific and measurable.
2. Have story points associated with it. Use 1, 2, 3 or 5 story points.
3. Be numbered and arranged in order of priority.

**Proposed Technical Setup**

You have to implement this project in .NET using a MySQL database and WPF.

**Lo-Fi User Interface**

What will the UI look like? Submit a **storyboard** using diagrams containing rough sketches of your product's user interface. These diagrams should depict the major UI used to implement the user stories you submit. For example, if one of your user stories was to allow a user to initiate a stock purchase, you should draw the initial UI that is presented when the user wishes to purchase a stock, along with any other major windows, messages, etc. that appear as the user navigates through this use case. The diagrams can be drawn by hand or computer. If a window leads to a dialog box, drop-down box, etc., include it as a sub-diagram. Your diagrams do not need to be pretty to get full credit, but they should be clear and legible. The main point is to illustrate your thoughts about what information should be shown and how the user will use the software.