

Projektantrag

Disclaimer

Bevor Sie den Projektantrag lesen, möchten wir Sie darauf aufmerksam machen, dass keiner von uns im betrieblichen Umfeld mit Python arbeitet und wir uns für dieses Projekt in die technischen Tiefen von Python begeben haben; und immer noch in der Einarbeitung sind. Dementsprechend fällt es uns schwer, zu beurteilen, inwiefern unser Vorhaben Ihren Vorgaben von einem Anteil in der Programmierung der Anwendung entspricht. Sollten Sie, nachdem Sie diesen Antrag gelesen haben, den Eindruck bekommen, dass unser Vorhaben nicht ausreichen wird: Kommen Sie gerne auf uns zu.

Beschreibung des Projekts

Das Projekt soll eine Anwendung sein, mit der es möglich ist, die eigenen Verkäufe und Käufe im Auktionshaus des MMORPGs World of Warcraft von Blizzard Activision zu festzuhalten. Die genannten Käufe und Verkäufe besitzen jeweils einen Preis, für den sie getätigt wurden, sowie eine ID und ein zugehöriges Item, welches gehandelt wird. Als Währung, in der die Items ge- oder verkauft werden, soll hier die WoW-Ingame-Währung genutzt werden, die sich aus Werten für Gold, Silber und Kupfer zusammensetzt: Gold ist der wertvollste Teil, gefolgt von Silber und abschließend Kupfer. Das Programm soll eine Auswertung der Käufe und Verkäufe ermöglichen, die in der GUI dargestellt wird, sodass der User einen Verlauf seiner Transaktionen nachvollziehen und analysieren kann.

Genutzte Hilfsmittel

Als Hilfsmittel für die Entwicklung der Rest-Schnittstelle soll das Framework "FastAPI" genutzt werden. Für die Erstellung der GUI wird "Tkinter" und für die Visualisierung der Daten "Matplotlib" herangezogen.

Erste Lösungsidee

Als eine mögliche Lösung wurde sich für eine REST-Schnittstelle entschieden, die zu implementieren ist. Diese Schnittstelle soll eingehende Http-Requests empfangen und verarbeiten können. Sei es hierbei die Rückgabe einer Auswertung oder die persistente Speicherung eines Kaufs/Verkaufs. Die persistente Speicherung der Handlungen soll in einer PostgreSQL Datenbank geschehen, welche isoliert in einem Docker Container mitgeliefert wird. Die Benutzerführung wird in der GUI mittels Tkinter ermöglicht. Die Visualisierung der Daten, die in der GUI dargestellt wird, erfolgt über die Einbindung von Matplotlib.

Für das Backend bzw die REST-Schnittstelle ist Marius Birk zuständig, während Marjan Schneider das Frontend mithilfe von "Tkinter" übernimmt.