Auto Buy bot

28.10.2020

BBB

Auto buy Bot

Metainformationen

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Autor** | **Bemerkung** |
| 28.10.2020 | 1.0 | T.Krebs | Erstellung des Dokumentations Grundgerüsts |
| 4.11.2020 | 1.1 | J.Meier | Eintragen von verschiedenen Daten (Tools, Anforderungsanalyse, Pflichtenheft, Tätigkeitsliste, Entscheidungsmatrix, Fazit) |
| 11.11.2020 | 1.1 | - | An der Dokumentation wurde nicht weitergearbeitet |
| 18.11.2020 | 1.2 | L.Emenegger, J.Meier, T.Krebs | Eintragen von verschiedenen Daten (Testspezifikation, Entscheidungsmatrix) |
| 25.11.2020 | 1.3 | L.Emenegger, J.Meier, T.Krebs | Eintragen von verschiedenen Daten (Testprotokoll, Testbericht, Selbstreflexion, Fazit, Zeitplan einfügen, Realisieren) |

Status

|  |  |  |  |
| --- | --- | --- | --- |
| **In Bearbeitung** | **In Bearbeitung** | **In Prüfung** | **Freigegeben** |
| 28.10.2020 | X |  |  |
| 4.11.2020 | X |  |  |
| 11.11.2020 | X |  |  |
| 18.11.2020 | X |  |  |
| 25.11.2020 | x | X | X |

Inhalt

[1 Informieren 4](#_Toc57210458)

[1.1 Projektantrag 4](#_Toc57210459)

[1.2 Projektantrag-E-Mail 5](#_Toc57210460)

[1.3 Idee 5](#_Toc57210461)

[1.4 Anforderungsanalyse 6](#_Toc57210462)

[1.5 Tools 6](#_Toc57210463)

[1.6 Zeitplan 7](#_Toc57210464)

[2 Planen 8](#_Toc57210465)

[2.1 Pflichtenheft 8](#_Toc57210466)

[2.2 Situation 8](#_Toc57210467)

[2.3 Ziele 8](#_Toc57210468)

[2.4 Anforderungen 8](#_Toc57210469)

[2.5 Tätigkeitsliste 9](#_Toc57210470)

[2.6 Testspezifikationen 10](#_Toc57210471)

[3 Entscheiden 13](#_Toc57210472)

[3.1 Entscheidungsmatrix 13](#_Toc57210473)

[3.2 Fazit 13](#_Toc57210474)

[4 Realisieren 14](#_Toc57210475)

[4.1 Visual Studio Code Download und Installation 14](#_Toc57210476)

[4.2 Code Review 15](#_Toc57210477)

[4.3 Fazit 17](#_Toc57210478)

[4.4 PyPy 17](#_Toc57210479)

[4.5 Module installieren 18](#_Toc57210480)

[4.6 Python installieren 18](#_Toc57210481)

[5 Kontrollieren 19](#_Toc57210482)

[5.1 Testprotokoll 19](#_Toc57210483)

[5.2 Testbericht 19](#_Toc57210484)

[6 Auswertung 20](#_Toc57210485)

[6.1 Selbstreflexion 20](#_Toc57210486)

[6.2 Fazit 23](#_Toc57210487)

# Informieren

## Projektantrag

|  |
| --- |
| **Projektname:** |
| Auto Buy Bot |

|  |
| --- |
| **Beschreibung:** |
| Wir wollen ein Bot programmieren, welcher ausgewählte Produkte (Klamotten, Sneakers usw.) automatisch in den Warenkorb legt.  Wir wollen, dass der Bot selbst auf die Website geht und wir in unserem Script angeben können, wie das Produkt heisst, welche Farbe es hat und unter welcher Kategorie es zu finden ist. Wenn der Bot das Produkt ausgewählt hat wollen wir, dass er das Produkt in den Warenkorb legt.  Dabei wollen wir neue Kenntnisse mit JSON, JavaScript und Python sammeln. Den Bot wollen wir hauptsächlich mit Python in Visual Studio Code programmieren.  Wir wollen den Bot so gestalten, dass er nicht mit simulierten Klicks arbeitet, sondern sich die ID des Eingabefeld holt und die Daten automatisch einfügt. Z.B. mit dem Command document.getElementByID(ProductName)  Ziele:   * Ein Bot zu programmieren, der ein ausgewähltes Produkt automatisch in den Warenkorb legt. * Grundlegendes wissen erlangen über Programmierung eines Bots. * Wissen wie ein Bot arbeitet/funktioniert. * Grundkenntnisse mit Python sammeln. |

|  |
| --- |
| **Module** |
|  |

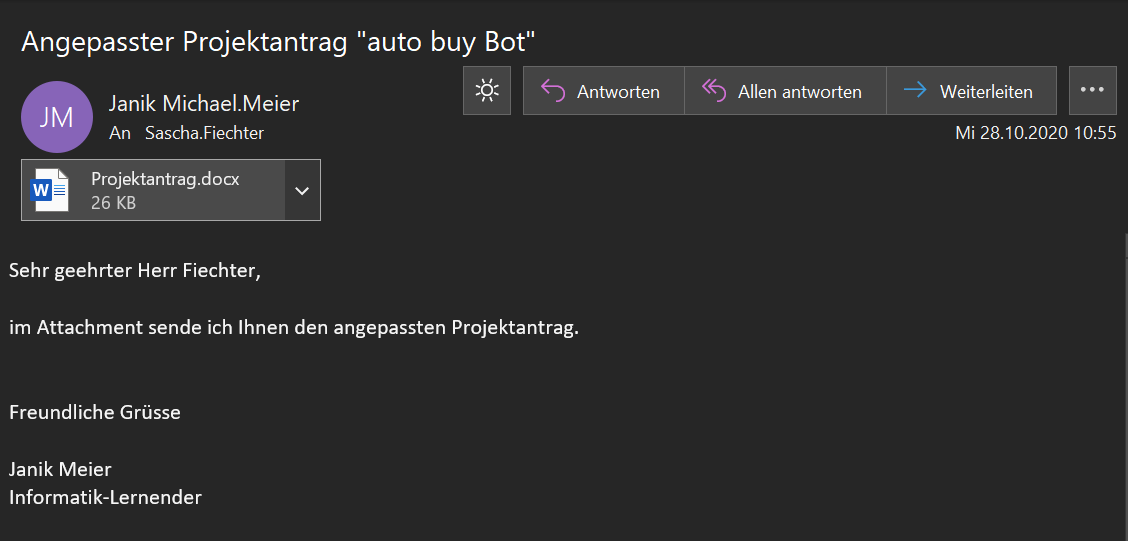
|  |
| --- |
| **Zusätzlich benötigtes Material:** |
| - |

|  |  |  |
| --- | --- | --- |
| **Auftraggeber:** | **Weitere Kontaktpersonen:** | **Begleitperson BBB:** |
| N/A | N/A | Sascha Fiechter |

|  |  |
| --- | --- |
| **Projektleiter:** | **Teammitglieder:** |
| Janik Meier | Timo Krebs, Lauren Emmenegger |

|  |  |
| --- | --- |
| **Termin Projektstart:** | **Termin Projektende:** |
| 28.10.2020 | 25.11.2020 |

## Projektantrag-E-Mail



## Idee

Nach der Suche nach einem neuen Projekt sind wir auf die Idee gestossen einen Product Buy Bot zu programmieren, um Markenprodukte die neu herauskommen möglichst schnell kaufen zu können. Der Bot soll ausgewählte Produkte (Klamotten, Sneakers usw.) automatisch in den Warenkorb legen. Janik hatte sich bereits informiert und wusste, dass es gut möglich war.

Nach kurzer Suche haben wir danach ein geeignetes Python Tutorial gefunden. Wir wollen nun mit Hilfe des Tutorials ein Bot mit Python programmieren, um Grundkenntnisse über Python zu erlangen und so viel zu programmieren, wie wir können.

## Anforderungsanalyse

|  |  |
| --- | --- |
| **Nr.** | **Anforderung** |
|  | Der Python Code ist in Visual Studio Code ausführbar. |
|  | Das Programm öffnet selbst die Webseite und navigiert zu dem gewünschten Produkt. |
|  | Man kann die Farbe des Produkts auswählen. |
|  | Man kann die Grösse des Produkts auswählen, wenn es ein Kleidungsstück ist. |
|  | Das Produkt wird automatisch in den Warenkorb gelegt und die Bestelldaten werden ausgefüllt. |

## Tools

|  |  |
| --- | --- |
| **Nr.** | **Tools** |
| 1 | Chrome Version 86.0.4240.183 (Offizieller Build) (64-Bit) |
| 2 | Visual Studio Code |
| 3 | Word |
| 4 | Excel |
| 5 | OneNote |
| 6 | SVN |
| 7 | Beautifulsoup 4 |
| 8 | Splinter |
| 9 | Request |
| 10 | Python |
| 11 | LXML |
| 12 | YouTube |
| 13 | Teams |
| 14 | SharePoint |

## Zeitplan



# Planen

## Pflichtenheft

Wir möchten einen Bot programmieren, welcher in Online-Shops Produkte in den Warenkorb legt. Der Bot sollte möglichst schnell sein und so zeitlich limitierte und begehrte Produkte beanspruchen können.

## Situation

Ausgangslage

Limitierte Produkte von z.B. Nike sind oft sehr schnell ausverkauft, da Bots diese sehr schnell kaufen können. Diese limitierten Artikel können dann teurer weiterverkauft werden. Wir möchten durch unseren Bot auch eine Chance darauf bekommen.

Ist-Zustand

Wir wissen ungefähr, wie der Bot funktioniert, kennen uns aber noch nicht so gut mit dessen Programmierung aus.

Soll-Zustand

Das Projekt soll bis am 25ten November fertig sein, heisst der Bot und die Dokumentation sind fertig und korrekt im SVN hinterlegt. Der Bot soll käufliche Objekte online automatisch in den Warenkorb legen können. Wir möchten den Bot mi Python programmieren.

## Ziele

* Einen funktionierenden Bot zu programmieren.
* Der Bot soll die Objekte in den Warenkorb legen.
* Eventuell soll der Bot die Objekte kaufen können.
* Wir kennen die grundlegenden Funktionen von Python und können Module verwenden.

## Anforderungen

Der Bot soll automatisch käufliche Objekte in den Warenkorb legen, sobald diese zum Kauf verfügbar sind. Der Ablauf soll schnell erfolgen, innerhalb von ein paar Sekunden.

## Tätigkeitsliste



## Testspezifikationen

**Nummer: 1**

**Getestete Anforderung:**

Der Python Code ist in Visual Studio Code ausführbar.

**Voraussetzung:**

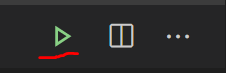
Pip3 ist installiert. Zudem müssen die Module Splinter, Request, Boutifulsoup4, Pylint mithilfe von Pip installiert sein. Man braucht auch noch den Geckodriver Version [geckodriver-v0.28.0-win64.zip](https://github.com/mozilla/geckodriver/releases/download/v0.28.0/geckodriver-v0.28.0-win64.zip) von GitHub <https://github.com/mozilla/geckodriver/releases>, die exe Datei muss im gleichen Ordner wie python.exe sein. Der Firefox Browser ist installiert. Visual Studio Code und Python 3.9.0 64-bit ist nach Anleitung von diesem Video installiert: Minute 0:33 bis 9:50

<https://www.youtube.com/watch?v=dNFgRUD2w68&t=491s>.

Die Datei «Auto\_Buy\_Bot.py» ist in Visual Studio Code geöffnet.

**Eingabe:**

In Visual Studio Code klickt man auf den hellgrünen Playbutton oben rechts, um das Programm zu starten.



**Ausgabe:** Es öffnet sich ein neues Fenster in Firefox.

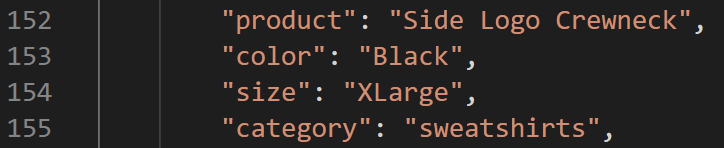
**Nummer: 2**

**Getestete Anforderung:**

Das Programm öffnet selbst die Webseite und navigiert zu dem gewünschten Produkt.

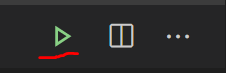
**Voraussetzung:**

alle Voraussetzungen aus Nummer 1, zudem müssen die Daten für ein Produkt auf Zeile 152-155 im Code hinterlegt sein.



**Eingabe:**

In Visual Studio Code klickt man auf den hellgrünen Playbutton oben rechts, um das Programm zu starten.



**Ausgabe:**

Es öffnet sich ein neues Firefox Fenster mit dem gewünschten Produkt.

**Nummer: 3**

**Getestete Anforderung:**

Man kann die Farbe des Produkts auswählen.

**Voraussetzung:**

alle Voraussetzungen aus Nummer 1 und die Erklärung zu den Values aus Nummer 5. Auf Zeile 153  zwischen den Anführungszeichen kann man die Farbe eingeben, bei diesem Foto ist die Farbe Black angegeben.

Values:



Zudem sollten die Values der Webseite mit den Values im Code übereinstimmen, wie im Video gezeigt. Ansonsten müssen diese Angepasst werden. In Englisch wäre das Value von «hinzufügen» «add to basket» und der Value von «zahlung durchführen» «process payment».

**Eingabe:**

Änderungen speichern und das Programm von Visual Studio Code starten.

**Ausgabe:**

Es öffnet sich ein neues Firefox Fenster, das Produkt wird in der gewünschten Farbe angezeigt, aber nur wenn die Farbe verfügbar ist.

**Nummer: 4**

**Getestete Anforderung:**

Man kann die Grösse des Produkts auswählen, wenn es ein Kleidungsstück ist.

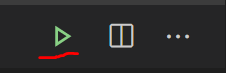
**Voraussetzung:**

alle Voraussetzungen aus Nummer 1.

Auf Zeile 154  zwischen den Anführungszeichen kann man die Grösse eingeben, bei diesem Foto ist die Grösse XLarge angegeben.

**Eingabe:**

Änderungen speichern (ctrl+s), in Visual Studio Code auf den hellgrünen Playbutton oben rechts klicken, um das Programm zu starten.



**Ausgabe:**

Es öffnet sich ein neues Firefox Fenster, das Produkt wird in der gewünschten Grösse und Farbe angezeigt, aber nur wenn die Grösse verfügbar ist.

**Nummer: 5**

**Getestete Anforderung:**

Das Produkt wird automatisch in den Warenkorb gelegt und die Bestelldaten werden ausgefüllt.

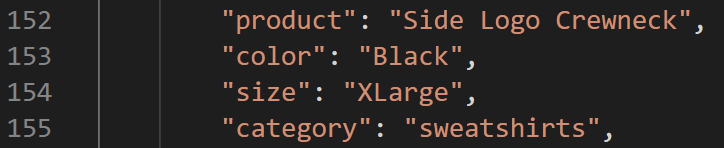
**Voraussetzung:**

alle Voraussetzungen aus Nummer 1-4.

Zudem müssen die Bestelldaten auf Zeile 156-167 ausgefüllt werden.

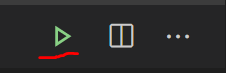


Die Produktdaten müssen auf Zeile 152-155 angegeben werden.



**Eingabe:**

In Visual Studio Code klickt man auf den hellgrünen Playbutton oben rechts, um das Programm zu starten.



**Ausgabe:**

Es öffnet sich ein neues Firefox Fenster, das Produkt wird in den Warenkorb gelegt, zudem werden die Bestellinformationen automatisch ausgefüllt und das Recaptcha erscheint auf dem Display.

# Entscheiden

## Entscheidungsmatrix

|  |  |
| --- | --- |
| **Python Pro:**   * viele Tutorials * grosser Libraryzugriff * mit Visual Studio Code Kompatibel * dynamisch geschrieben   **Contra:**   * langsamer * schwerer zum installieren | **C# Pro:**   * schneller * einfach zu installieren * höhere Leistung * Mit Visual Studio Code kompatibel   **Contra:**   * weniger Library Zugriff * statisch geschrieben * schwerer um Code zu schreiben |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | **Lösungskonzepte** | | | |
| **Auto buy Bot** | | | Python | | C++ | |
| **Anforderungen** | | **Gewichtung** | **Max = 10** | **=** | **Max =10** | **=** |
|  |  | **Nutzwert** |  | **Nutzwert** |
| Simplizität |  | **7** | 8 | **56** | 5 | **35** |
| Tutorials |  | **9** | 8 | **72** | 5 | **45** |
| Eigene Interessen | | **8** | 10 | **80** | 8 | **64** |
| Libraryzugriff | | **7** | 10 | **70** | 6 | **42** |
| Geschwindigkeit | | **10** | 6 | **60** | 10 | **100** |
| Mit Visual Studio Code kompatibel | | **8** | 10 | **80** | 10 | **80** |
| **TOTAL** | |  |  | **418** |  | **366** |

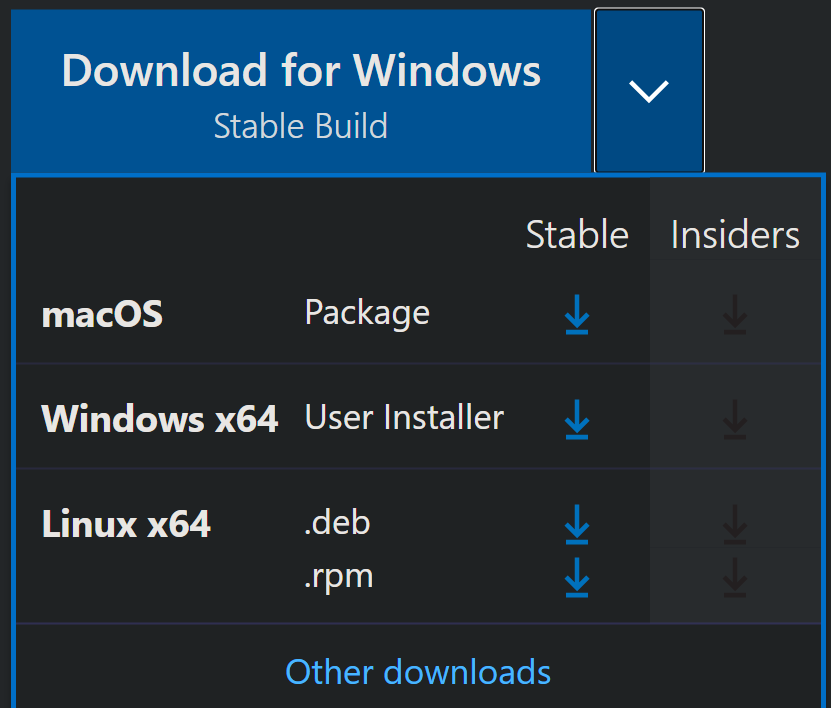
## Fazit

Die Endgültige Entscheidung fiel auf Python. Wir hatten bereits ein sehr gutes Tutorial dazu gefunden und unser Interesse daran war ebenfalls grösser. Die Programmiersprache war zwar langsamer was wir sehr schade fanden, jedoch war es wichtiger für uns eine Programmiersprache zu verwenden welche wir dann auch besser verstehen und der Code auch funktioniert.

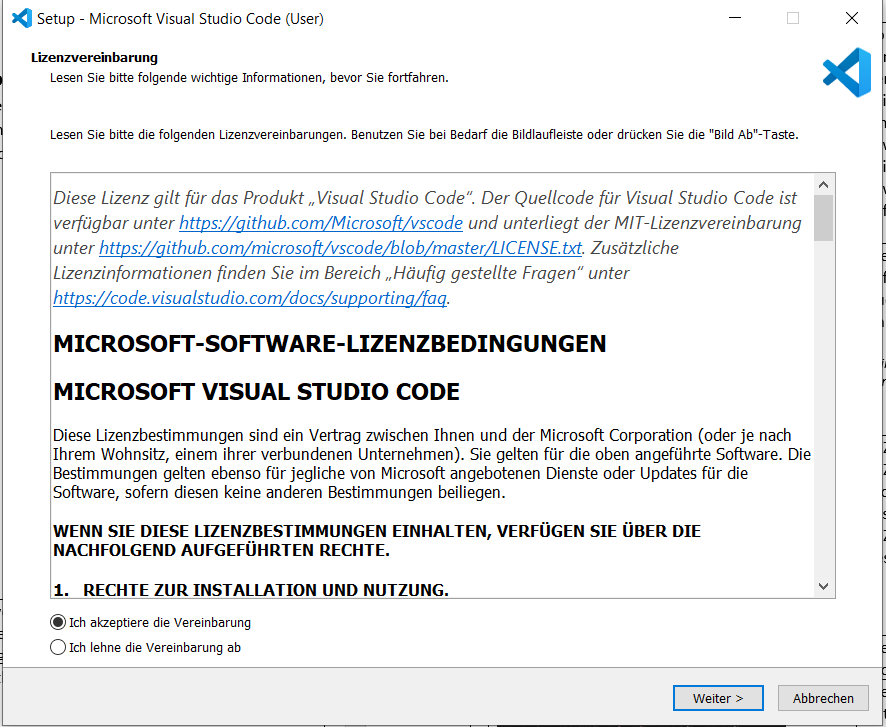
# Realisieren

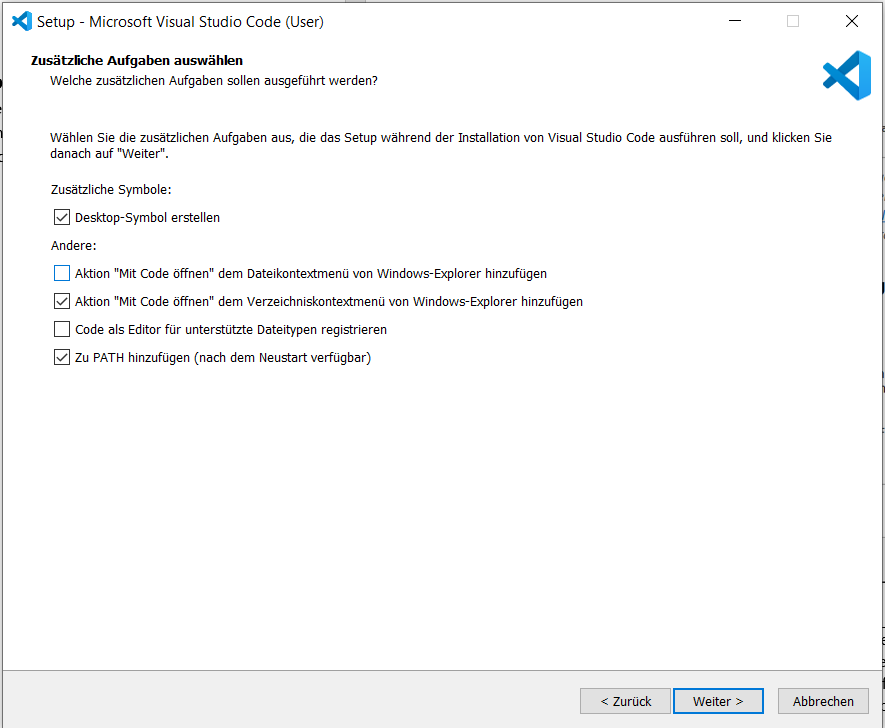
## Visual Studio Code Download und Installation

Um mit Python programmieren zu brauchten wir als erstes einen Code Editor. Wir habe uns für Visual Studio Code entschieden, weil wir bereits Erfahrung damit hatten. Auf [der Visual Studio Code Website](https://code.visualstudio.com/) konnten wir einen Installer Herunterladen für den Editor.



Als nächstes haben wir die heruntergeladene Datei VSCodeUserSetup-x64-1.51.1.exe ausgeführt. Wir konnten VSCode mit folgenden Einstellungen installieren:

`



Bei dem Nächsten Fenster mussten wir nur noch auf Installieren drücken und der Editor war heruntergeladen.

## Code Review

|  |  |  |
| --- | --- | --- |
| 1 |  | Hier werden die Codebibliotheken importiert, welche im Code verwendet werden. Um z.B. zufällige Zahlen zu generieren, benötigt man die Bibliothek «random», Zeile 7. |
| 2 |  | Hier kann man den Webseitenlink eingeben. In Zeile 17 wird auf die Info Liste (Tabellenspalte 8) verwiesen. Zeile 19, 20 verweisen wir auf den Firefox Browser. |
| 3 |  | Von Zeile 23-29 versucht der Bot, den Quelltext von der oben angegebenen Webseite herunterzuladen. Zeile 34,35 speichert er alle Links im Quelltext in eine Liste. Einer dieser Links führt zum gewünschten Produkt. Zeile 42-47 speichert die Ähnlichkeit mit dem gewünschten Produktnamen mit jedem gefundenen Produktnamen in einer Liste. |
| 4 |  | Hier werden die Ähnlichkeiten zu den gefundenen Produktnamen + Links hinzugefügt. Auch für gefundene Farben werden Ähnlichkeiten zugewiesen. Eine solche Liste sieht so aus: *[[Link, Produktname, Ähnlichkeit], [Link, Produktname, Ähnlichkeit], …]* |
| 5 |  | In Zeile 80-87 werden leere Listen entfernt.  In Zeile 89-98 werden die fünf ähnlichsten Produktnamen und -farben in eine Liste gespeichert. In Zeile 101-105 wird der richtige Produktlink bestimmt. |
| 6 |  | Hier wird die Produktwebseite besucht und das Produkt in den Warenkorb hinzugefügt. |

|  |  |  |
| --- | --- | --- |
| 7 |  | Hier werden Zahlungsinformationen und Adresse automatisch ausgefüllt und auf «Zahlung durchführen» geklickt. |
| 8 |  | In dieser Liste muss man seine Zahlungsinformationen, Adresse und Produktinformationen eintragen. |
| 9 |  | Hier wird der Bot ausgeführt. In Zeile 168-172 wird jede Sekunde eine Anfrage gesendet, ob das Produkt online verfügbar ist. Es sendet maximal 100 Anfragen, danach wird eine Fehlermeldung ausgegeben. Wenn es das Produkt findet, wird es in den Warenkorb gelegt und die Zahlungsinformationen ausgefüllt. |

## Fazit

Der Bot findet anhand eines Produktnamens und -farbe das Produkt, für das die Angaben am besten zutreffen. Ist ein Produkt gefunden, wird es in den Warenkorb gelegt und die Zahlungsinformationen werden ausgefüllt. Der Bot kann Produkte kaufen, die nach dem Release sehr schnell ausverkauft sind.

## PyPy

Um den Bot schneller zu machen wollten wir die Programmiersprache PyPy benutzen. PyPy ist ein Just-in-time-Compiler für Python welche 4-Mal so schnell sein sollte. Wir haben unter der Download Website von [PyPy](https://www.pypy.org/download.html) den Ordner welcher PyPy beinhaltet heruntergeladen. Wir haben die Version PyPy3.7 für Windows 32 Bit heruntergeladen.

Wir waren zuerst verunsichert das es keine 64 Bit Version hatte wurden jedoch informiert, dass es so auch funktionieren sollte. Wir haben sehr lange nach Informationen gesucht wie man PyPy installieren muss und sind auf dieses [Video](https://www.youtube.com/watch?v=7XeDbMuMy60) gestossen welches uns von Minute 0-3:55 den Download und die Installation erklärt.

Um den Bot benutzen zu können mussten wir noch Pip3 installieren, um Module für Python zu installieren. Dies haben wir mit der Eingabeaufforderung (CMD) gemacht. Wir haben mit mehreren Befehlen versucht Pip3 zu installieren, sowie die restlichen Module. Wir haben es leider nicht korrekt geschafft und mussten weiterhin Python benutzen.

## Module installieren

Python-Module installiert man über das CMD mit dem Pip Befehl.   
Wir haben die Module mit diesen Befehlen installiert:

* python -m pip install requests
* pip3 install bs4
* pip3 install splinter

## Python installieren

Python haben wir von der Website [Python.org](https://www.python.org/downloads/) heruntergeladen und installiert. Dabei haben wir es zum PATH hinzugefügt. Den Skriptordner mussten wir auch zum PATH hinzufügen. Für Visual Studio Code mussten wir ausserdem eine Python Extension installieren.

# Kontrollieren

## Testprotokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nummer | Resultat | Datum | Tester | Unterschrift |
| 1 | Funktioniert | 25.11.2020 | T. Krebs |  |
| 2 | Funktioniert | 25.11.2020 | T. Krebs |  |
| 3 | Funktioniert | 25.11.2020 | T. Krebs |  |
| 4 | Funktioniert | 25.11.2020 | T. Krebs |  |
| 5 | Funktioniert | 25.11.2020 | T. Krebs |  |

## Testbericht

Das Produkt kann angegeben werden und wird danach mit den korrekten Angaben in Warenkorb gelegt, die Bestellinformationen werden automatisch ausgefüllt und der Zahlungsknopf wird automatisch betätigt. Alle Anforderungen wurden bestanden. Probleme können auftreten, wenn man die Kategorie nicht korrekt eingibt oder das Produkt und die Farbe mit sehr vielen Fehlern eingibt. Den Bot kann man anwenden ist jedoch langsamer im Vergleich zu professionellen Bots, die bis zu 10'000 CHF kosten können.

# Auswertung

## Selbstreflexion

**Timo Krebs:**

Mit meiner Leistung und meinem Beitrag zu dem Projekt bin ich im Großen und Ganzen zufrieden.

Als erstes wurde ich beauftragt ein geeignetes Tutorial und genügend Informationen zu finden, um den Bot programmieren zu können. Ich habe nach einiger Zeit Suche ein geeignetes Tutorial finden, welches sehr zu dem Projekt beigetragen hat und allen sehr geholfen hat.

Zu beginn konnte ich viel zu dem Bot beitragen und habe neue Ideen hineingebracht, jedoch konnte ich diese nicht selbst umsetzen. An dem Punkt wo wir nur noch neue Informationen finden konnten welche auf einer Website waren und der Befehl nur ganz kurz beschrieben wurden konnte ich nicht mehr weitermachen und war überfordert, was mich sehr genervt hat da ich ab diesem Punkt für die Weiterentwicklung des Bots nutzlos war.

Was ich gut gemacht habe:

* Ich konnte viel Informationen beschaffen und ein geeignetes Tutorial finde für den Bot
* Ich konnte neue Ideen einbringen, um den Bot zu verbessern

Was ich noch besser machen könnte:

* Mich mit den komplizierteren Themen länger und intensiver beschäftigen
* Mehr an der Dokumentation arbeiten, wenn ich so sehr festgehangen bin, dass ich nichts mehr machen konnte

**Lauren Emmenegger:**

Ich bin hauptsächlich zufrieden mit meiner Arbeit.  
Am Anfang habe ich nur abgeschrieben, was im Tutorial erklärt wurde. Mit der Zeit erkannte ich, was die verschiedenen Codeabschnitte bewirken. Danach konnte ich den Code modifizieren und eigene Funktionalitäten einbauen.

Ich war manchmal abgelenkt und nicht sehr produktiv, doch gegen Ende konnte ich noch das Programm abschliessen. Teilweise sass ich eine Stunde am Code und suchte nach Wegen, ein Ziel zu erreichen, und am Schluss kam ich nicht weiter, als ich schon war. Als ich entdeckte, wie ich zwei Listen auf eine bestimmte Art und Weise zusammenfügen kann, lief es wieder schneller voran mit der Entwicklung.

Was ich gut gemacht habe:

* Entwicklung neuartiger Technologie für den Bot
* Ambitioniertes und motiviertes Arbeiten

Was ich verbessern kann:

* Mich weniger ablenken lassen

**Janik Meier:**

Ich bin mit dem Projekt zufrieden, da wir als Endprodukt einen funktionierenden Bot haben. An den ersten zwei Projekttagen hatten wir etwas Schwierigkeiten, da Timo im Home-Office war. Dadurch mussten wir ständig in einem Call sein, was nicht so gut war, da die Motivation manchmal darunter gelitten hat. An den restlichen Projekttagen haben wir als Gruppe gut zusammengearbeitet und uns gegenseitig unterstützt. Zudem sind wir dann schnell vorangekommen.

Mit meiner Arbeit bin ich auch zufrieden, da ich den roten Faden nie verloren habe. Und mein Team unterstützen konnte. Während der Dokumentation am letzten Projekttag war ich teilweise nicht sehr produktiv, da ich fast keine Motivation hatte.

Was ich gut gemacht habe:

* Ich konnte mein Team unterstützen.
* Ich konnte immer den Überblick behalten.

Was ich verbessern kann:

* Motivierter und Konzentrierter an der Dokumentation arbeiten
* Mein Team mehr Motivieren

## Fazit

Wir konnten die 5 Wochen, die wir für das Projekt erhalten haben, immer effizient nutzen. Am Anfang konnten wir uns einen guten Überblick verschaffen, Informationen beschaffen und ein gutes Grundgerüst für die Dokumentation erstellen.

Das ganze Projekt hat uns sehr viel Spass gemacht da wir neue Kenntnisse in Python erlangen konnten und das Thema uns alle sehr interessiert hat. Unser am Anfang gestecktes Ziel haben wir schon am zweiten Projekttag erreichen können, dadurch hatten wir noch reichlich Zeit den Bot nach unseren Interessen auszuweiten.

Jeder hat sehr viel Engagement gezeigt und war motiviert. Über das ganze Projekt hatten wir sehr gute Teamarbeit. Den Bot haben wir bereits benutzt, konnten aber noch keine Erfolge feiern.

Wir haben ebenfalls versucht den Bot mit PyPy zu benutzen dies hat jedoch nicht funktioniert was uns sehr enttäuscht hat. Wir konnten die Probleme, die aufgetaucht sind mit dem ganzen Team lösen und beheben. Ich denke wir können im nächsten Projekt nicht viel besser machen, jedoch war das Homeoffice an den ersten zwei Projekttagen eine Hürde, die wir meistern mussten. Das nächste Mal sind wir dann besser darauf vorbereitet und wissen auf was man achten muss.