

Pflichtenheft

Thema des Auftrags: QR-Generator/Scanner

Jahrgang: 4AHIT

Schuljahr: 2022/23

Projektteam

Name	Rolle	Username	E-Mail
Lazău Larissa	Frontend, Backend	lazau190144	lazau190144@s r.htlweiz.at
Postl Denise	Frontend, Backend	postl190070	postl190070@s r.htlweiz.at

Ansprechpartner

Rolle	Name	Telefon	E-Mail
Auftraggeber	Prof. Ebner		

Versionskonen

Ver.	Datum	Autor(en)	Änderungsgrund / Bemerkungen	git-tag
0.1		KZ1, KZ2	Ersterstellung als Diskussionsvorlage	
0.2		KZ1	Erste Revision	
1.0		KZ2	Endversion als Diskussionsgrundlage	
1.1		KZ1, KZ2	Endversion	

Beglaubigung

Datum	Kunde	Datum	Team

Inhaltsverzeichnis

Projektteam	1
Ansprechpartner	1
Versionskonen.....	1
Beglaubigung.....	1
Inhaltsverzeichnis	2
1. Allgemeines - Zweck und Ziel dieses Dokuments	3
1.1. Untersuchungsanliegen der individuellen Themenstellung.....	3
1.2. Zielsetzung	3
1.3. Geplantes Ergebnis	3
1.4. Projektbezug.....	3
2. Zielkriterien	4
2.1. Hardwarespezifikation	4
2.1.1. Human Machine Interface (HMI).....	4
2.2. Softwarespezifikation.....	4
2.2.1. Human Machine Interface (HMI).....	4
2.3. Test und Industrialisierung.....	4
3. Erweiterte Ziel- oder Wunschkriterien.....	5
4. Geräteaufbau.....	6
4.1. Blockschaltbild der Hardware	6
4.2. Struktogramm der Software.....	6
5. Aufwände.....	7
5.1. Zeitliche Meilensteine in der Projektabwicklung	7
5.2. Finanzielle Aufwendungen.....	7

1. Allgemeines - Zweck und Ziel dieses Dokuments

In diesem Dokument wird festgehalten, welche Arbeiten im Rahmen des Projekts zu erledigen sind. Dazu werden konkrete Fälle explizit ein- oder ausgeschlossen. Zur besseren Übersicht befindet sich im Anhang eine stichwortartige Aufzählung der zu erledigenden Arbeiten. Die Anforderungen sind dabei in Muss- und Optionale Punkte unterteilt.

Auf die Implementierung wird im Pflichtenheft nicht eingegangen.

1.1. Untersuchungsanliegen der individuellen Themenstellung

Es soll ein QR-Generator für eine Paketmanager erstellt werden. Ein Benutzer kann Informationen für das Paket eingeben und ein QR-Code wird generiert. Weiters ist ein QR-Scanner zu erstellen, mit dem der Benutzer den QR-Code scannen kann und Informationen bezüglich des Pakets lesen kann.

1.2. Geplantes Ergebnis

Das geplante Ergebnis wäre ein funktionsfähiger QR-Generator und QR-Scanner. Ein Benutzer kann aus Informationen einen QR-Code generieren und aus einem bereits erstellten QR-Code kann er Informationen lesen. Der QR-Code wird generiert, ein PDF wird heruntergeladen und mittels Screenshots des QR-Codes im PDF können Informationen gescannt werden.

1.3. Projektbezug

Die Website ist grundsätzlich für den schulinternen Lernzweck gedacht aber auch für jede Person, die sich für das Erstellen und Scannen von QR-Codes interessiert. Prinzipiell sind der QR-Generator und der QR-Scanner Teil eines Projekts, welches das Sortieren von Paketen erleichtern soll.

2. Zielkriterien

2.1. Hardwarespezifikation

Hier wird beschrieben, wie die Hardware aufgebaut sein soll, beziehungsweise was das Gerät hardwaretechnisch können soll!

- PC, Notebook
- Ein QR-Code wird generiert und als PDF heruntergeladen. Dieser QR-Code kann aus einem PDF per Screenshot gescannt werden. Somit können QR-Codes auch auf PC's ohne Kamera gelesen werden.

2.1.1. Human Machine Interface (HMI)

Eine Beschreibung, was der Nutzer manuell am Gerät machen kann.

- Der User kann Informationen im QR-Generator eingeben
- Nach Bestätigen wird ein PDF mit dem QR-Code heruntergeladen
- Den QR-Code des PDF's kann der User später einscannen – mittels Screenshot werden Informationen gescannt - und seine eingegebenen Informationen werden ausgegeben

2.2. Softwarespezifikation

Hier wird beschrieben, wie die Software aufgebaut sein soll, beziehungsweise was das Gerät softwaretechnisch können soll!

- Python

- Flask
- SQLite

Das Programm des QR-Code-Generators nutzt das Python-Webframework Flask, um eine Webseite zu erstellen, auf der QR-Codes generiert werden können und als PDF-Datei heruntergeladen werden. Dazu wird eine SQLite-Datenbank verwendet, um Kundendaten, Produktinformationen, Standortinformationen und QR-Code-Daten zu speichern.

Das Programm des QR-Code-Generators besteht aus einem Flask-Server, der auf die Eingabe von Formulardaten durch den Benutzer reagiert, die Daten in die Datenbank einfügt, einen QR-Code generiert und eine PDF-Datei mit dem QR-Code erstellt. Die PDF-Datei wird dann heruntergeladen.

Der Code des QR-Code Scanners nutzt ebenfalls Flask, ein Webanwendungs-Framework, um eine Anwendung zu erstellen, die einen Screenshot des Bildschirms aufnimmt, nach QR-Codes sucht und die Daten in einer SQLite-Datenbank abrufen. Wenn ein QR-Code gefunden wird, werden die dazugehörigen Datensätze aus der Datenbank abgefragt und auf einer Ergebnisseite angezeigt. Andernfalls wird eine Fehlermeldung angezeigt.

2.2.1. Human Machine Interface (HMI)

Eine Beschreibung der Möglichkeiten zur Beeinflussung des Betriebs des Gerätes.

- QR-Code generieren und Downloaden
- QR-Code per Screenshot scannen und Informationen erhalten

2.3. Test und Industrialisierung

Beschreibung, wie die Funktion des Gerätes oder der Anlage über den geforderten Arbeitsbereich sichergestellt werden kann.

Die Informationen der eingegebenen Daten für einen QR-Code werden in einer Datenbank gespeichert. Diese Informationen werden nach Scannen eines bereits erstellten QR-Codes wieder aus der Datenbank gelesen. Somit sollten die Informationen keine unerwünschten Änderungen enthalten.

3. Erweiterte Ziel- oder Wunschkriterien

Beschreibung was über die Normalanforderung an das Projekt noch realisiert werden könnte.

Gewünscht wäre eine funktionsfähige Applikation, wo der User einerseits QR-Codes generieren kann, welche dann nach vollständigem Ausfüllen des Formulars als PDF gedownloadet werden. Von einem PDF kann dann ein Screenshot des QR-Codes gemacht werden, welcher dann gescannt wird und die Informationen, die sich hinter dem QR-Code befinden werden anschließend ausgegeben.

4. Geräteaufbau

4.1. Struktogramm der Software

Hier wird beschrieben, wie die Software arbeiten soll! Am besten ist die graphische Darstellung mittels Struktogramms oder Flussdiagramm. Auch die Darstellung eines Zustandsdiagrammes kann sinnvoll sein.

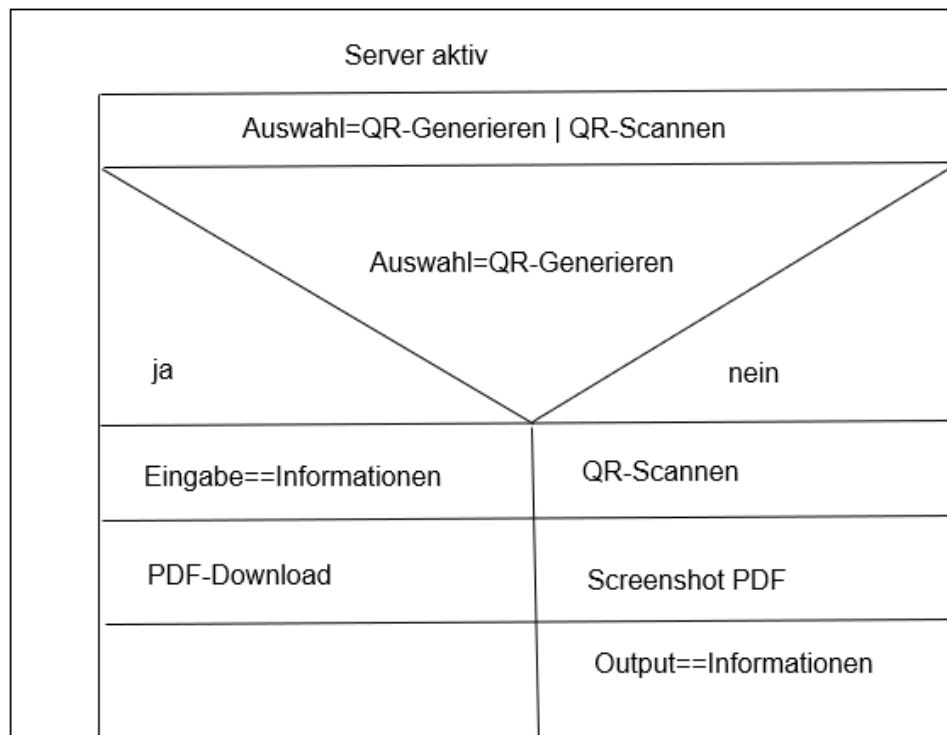


Bild 1: Struktogramm der Software

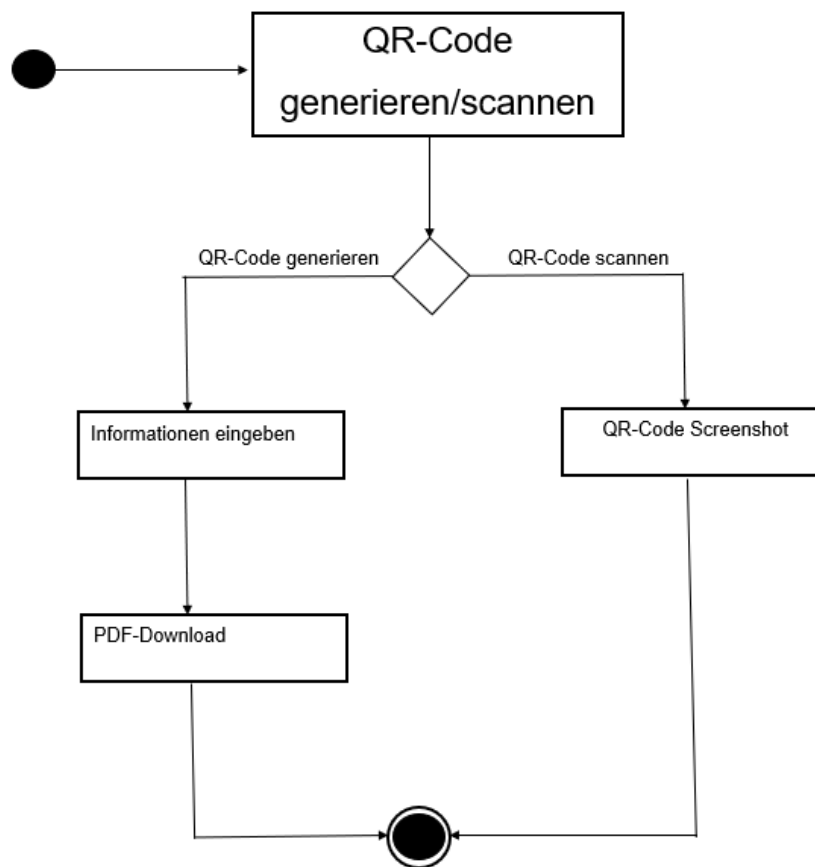


Bild 2: Zustandsdiagramm

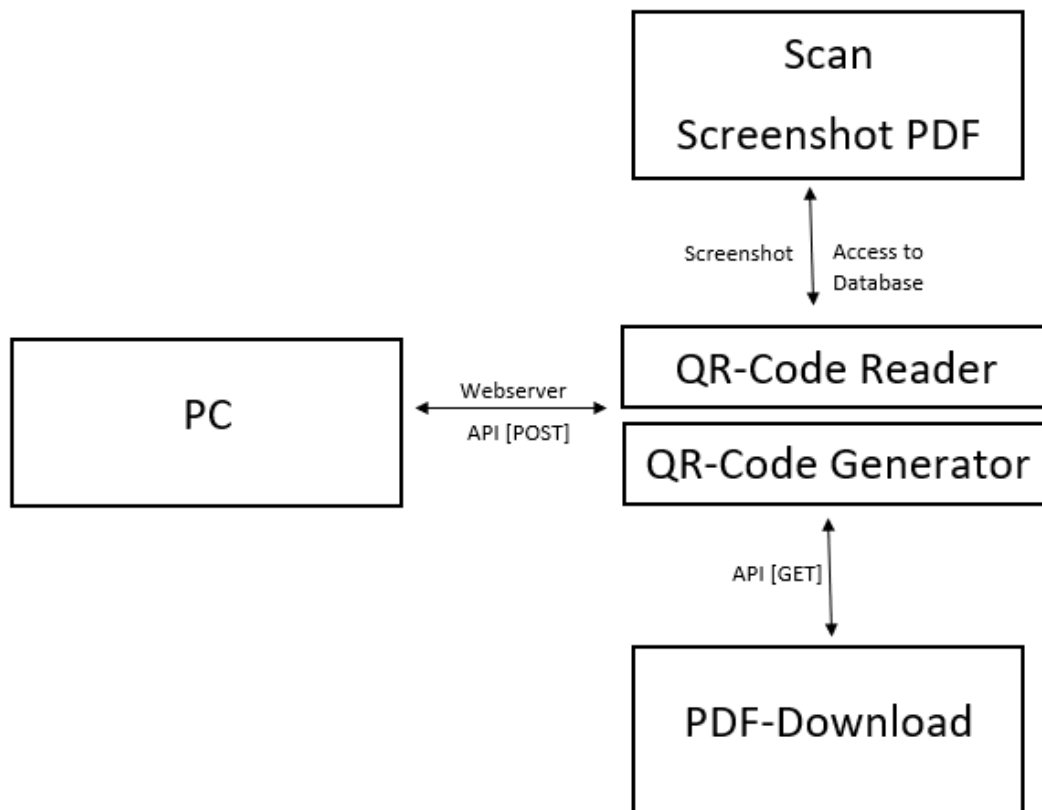


Bild 3: Beschreibung des Human Machine Interfaces

5. Aufwände

5.1. Zeitliche Meilensteine in der Projektabwicklung

Abschätzung der zeitlichen Durchführung anhand eines einfachen tabellarischen Projektablaufplanes. Dabei soll für jeden Diplomanden eine Tabelle erstellt werden.

5.1.1.

Datum	Meilenstein
09.03.2023	Fertigstellung der Planung des QR-Scanners u. Generators
23.03.2023	Fertigstellung des QR-Generators; Download des QR-Codes als PDF möglich
30.03.2023	Fertigstellung des QR-Scanners; Abschließen des Projektes einschließlich Dokumentation

5.2. Finanzielle Aufwendungen

Auflistung der HW und SW-Komponenten mit erster Abschätzung der Kosten.

Pos.	Beschreibung	Bezugsquelle	Kosten
1	Programmieren des QR-Scanners	Pfeifer, R. (2018). Stundensätze in der IT. IT-Alliance. Abgerufen am 27.03.2023, von https://www.it-alliance.at/2018/02/stundensaetze-in-der-it/	420 €
2	Programmieren des QR-Generators	Pfeifer, R. (2018). Stundensätze in der IT. IT-Alliance. Abgerufen am 27.03.2023, von https://www.it-alliance.at/2018/02/stundensaetze-in-der-it/	840 €

Die Kosten wurden folgendermaßen abgeschätzt:

- investierte Stunden x durchschnittlicher Stundensatz

angenommener durchschnittlicher Stundensatz: 70 €

investierte Stunden QR-Scanner: 6 h

investierte Stunden QR-Generator: 12 h

Abbildungsverzeichnis

Bild 1: Blockschaltbild der Hardware **Fehler! Textmarke nicht definiert.**

Bild 2: Detailschaltbild der Hardware **Fehler! Textmarke nicht definiert.**

Bild 3: Beschreibung des Human Machine Interfaces **Fehler! Textmarke nicht definiert.**

Bild 4: Struktogramm der Software

Bild 5: Zustandsdiagramm

Bild 6: Beschreibung des Human Machine Interfaces