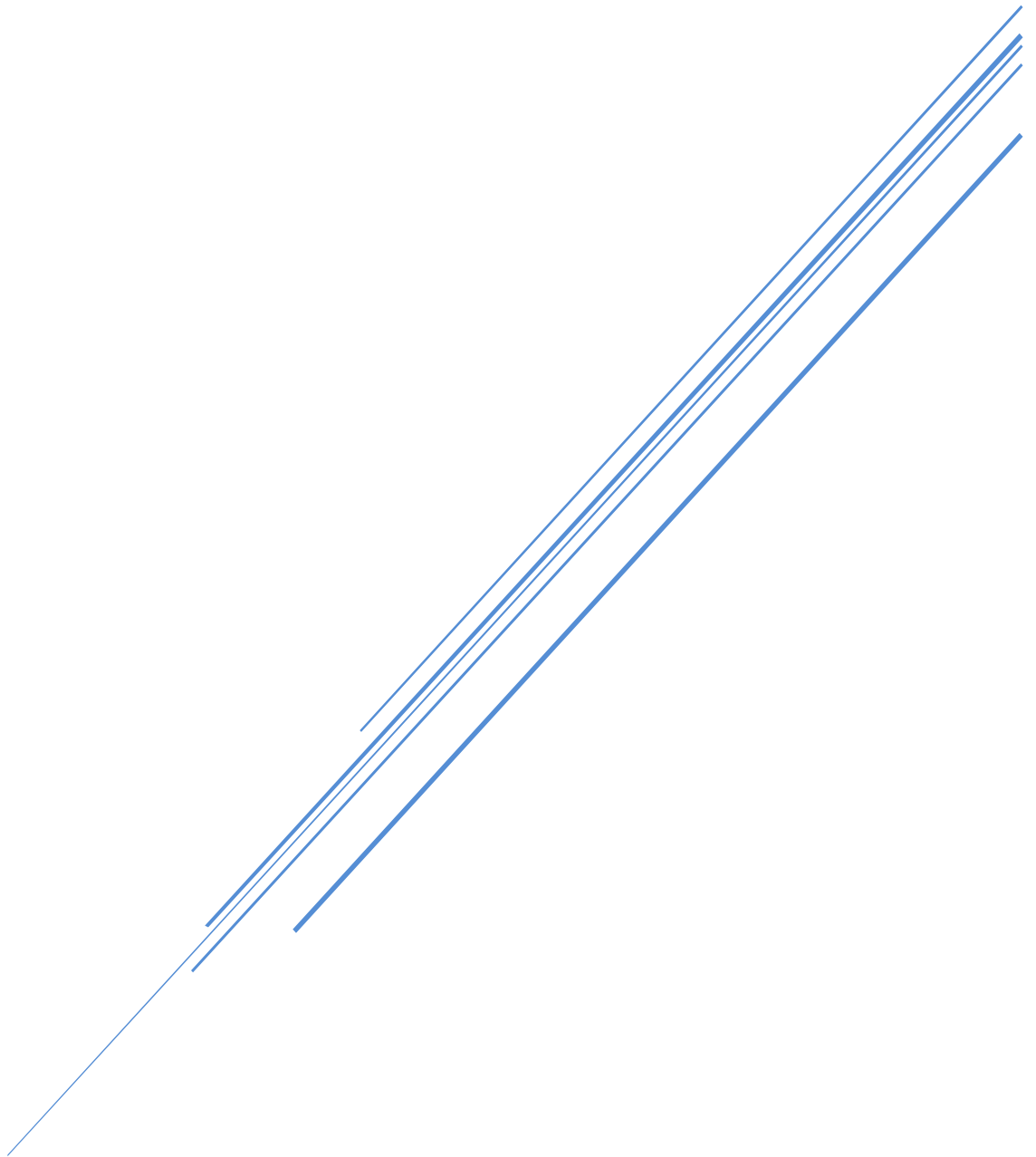


MESSEPROJEKT

Erik Sulzbach, Julian Beck, Julian Schönfeld, Kenrick Dehli



IT.Schule Stuttgart
SAE – ITS

Inhaltsverzeichnis

1.	Einführung	2
2.	Anforderungsanalyse	2
2.1	Situationsbeschreibung	2
2.1.1	Teilprojekt SAE: Datenerfassung Neukunden	2
2.1.2	Teilprojekt ITS: WLAN	2
3.	Teilprojekt ITS	2
3.1	Netzwerkinfrastruktur Stand	2
3.1.1	Netzwerkaufbau	2
3.1.2	Anbindung Messenetzwerk	4
3.1.3	Netzwerk Einrichtung und IP-Zuweisung	4
3.1.4	Routing	5
3.2	WLAN	6
3.2.1	Zugang und Sicherheit	6
3.2.2	Anbindung von Clients	6
3.3	Inbetriebnahme	6
4.	Teilprojekt SAE	7
4.1	Datenbank	7
4.1.1	Datenbankmodell z.B. Relationen-Modell	7
4.2	Aufbau und Funktionsweise	8
4.2.1	Architektur	8
4.2.2	USE Case und UML Diagramme	9
4.2.3	Prerequisites: Bibliotheken und Komponenten	11
4.2.4	Inbetriebnahme vor Ort	12
4.2.5	Technische Beschreibung der WebCam-Anbindung	12
4.2.6	Anleitung Bedienung durch den Kunden	14
4.2.7	Anleitung Datenabruf und Übermittlung	16
4.2.8	Testszenarien	17
	Anhang	18
	Verweise	19

1. Einführung

Im Rahmen des Projekts Messeauftritt soll eine WLAN-Infrastruktur und eine Software zur Erfassung von Kundendaten bereitgestellt werden.

2. Anforderungsanalyse

2.1 Situationsbeschreibung

Die Firma Juker plant den Besuch einer Messe. Auf der Messe sollen neben den üblichen Tätigkeiten nach Möglichkeit auch Daten potenzieller Neukunden erhoben und gespeichert werden. Zu diesem Zweck kann der Messestand Gutscheine ausstellen, mit denen vergünstigte Angebote auf der Messe wahrgenommen werden können. Voraussetzung ist die Registrierung im Portal der Firma Juker.

2.1.1 Teilprojekt SAE: Datenerfassung Neukunden

Während des Messeauftritts sollen von Kunden im Self-Service Kundenkarten erstellt werden können, mit denen dann der Zugang zu weiteren Messeangeboten möglich wird. Dabei sollen Nachname, Vorname, Anschrift und ein Bild erfasst werden. Zusätzlich sollen ein oder mehrere Produktgruppen angegeben werden können, für die besonderes Interesse besteht. Bei Firmenvertretern soll zusätzlich ein Datensatz für die Firma angelegt werden.

Die Speicherung der Daten soll langfristig in einer Datenbank erfolgen. Da die Zuverlässigkeit der Netzwerkverbindung während des Messeauftritts nicht immer sichergestellt werden kann, muss das Erfassungssystem auch offline funktionieren und in der Lage sein, die Daten auf Wunsch an die Firmenzentrale zu übermitteln.

Die gespeicherten Daten sollen von den MitarbeiterInnen auch abgerufen und durchsucht werden können. Da es sich um einen Self-Service handelt muss sichergestellt werden, dass nicht jede Person das System frei nutzen kann.

Für die Erfassung des Fotos soll eine Webcam angebunden werden.

2.1.2 Teilprojekt ITS: WLAN

Sie sollen für den Messeauftritt ein WLAN planen, da Sie nicht auf das dort verfügbare öffentliche WLAN zugreifen wollen. Zu diesem Zweck erhalten Sie vom Messeveranstalter einen LAN-Zugang mit einem eigenen Subnetz. Das WLAN soll nicht öffentlich sein und eine /26 Subnetmaske haben.

Das Netzwerk muss so aufgebaut sein, dass die im Teilprojekt SAE erfassten Daten bei Bedarf an die Firmenzentrale übermittelt werden können. Die Nutzung des Netzwerks soll nur für berechnigte Personen möglich sein. Sollte es bei Ihrem Ansatz notwendig sein, dass Besucher sich in das von Ihnen angebotene WLAN einwählen, dürfen MitarbeiterInnen und Besucher sich nicht im gleichen WLAN befinden.

3. Teilprojekt ITS

3.1 Netzwerkinfrastruktur Stand

3.1.1 Netzwerkaufbau

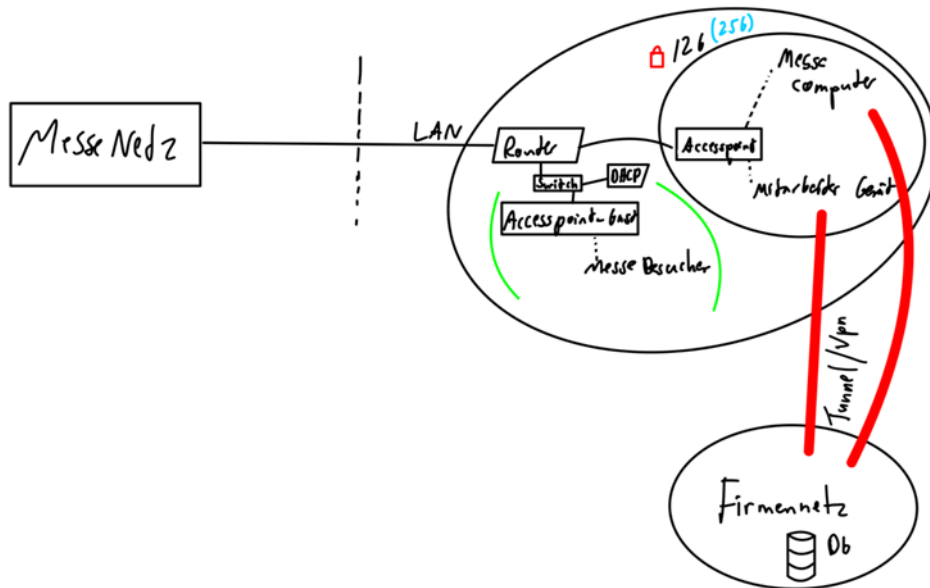
Das Messenetz wird zwei separate Netze haben, um die Administratoren von den Gästen, die sich registrieren wollen zu trennen. An dem Messestand wird es einen eigenen Router geben, der ins Messenetz routet. So unterscheiden wir zwischen dem Messenetz (Der Zugriff nach Außen) und dem Standnetz, welches eine Untergliederung in Admin- und Gästenetz vorweist.

Wie in der Abbildung zu erkennen, soll auf eine Datenbank zugegriffen werden, welche sich

außerhalb der Messe befindet. Dabei greifen die Administratoren über einen VPN auf die Datenbank zu. Den Administratoren wird ein eigener Accesspoint zur Verfügung stehen. Dieser soll mit WPA2 verschlüsselt werden. Administratoren PC bekommen feste IP-Adressen zugewiesen. Das Netz verfügt über keinen DHCP Server.

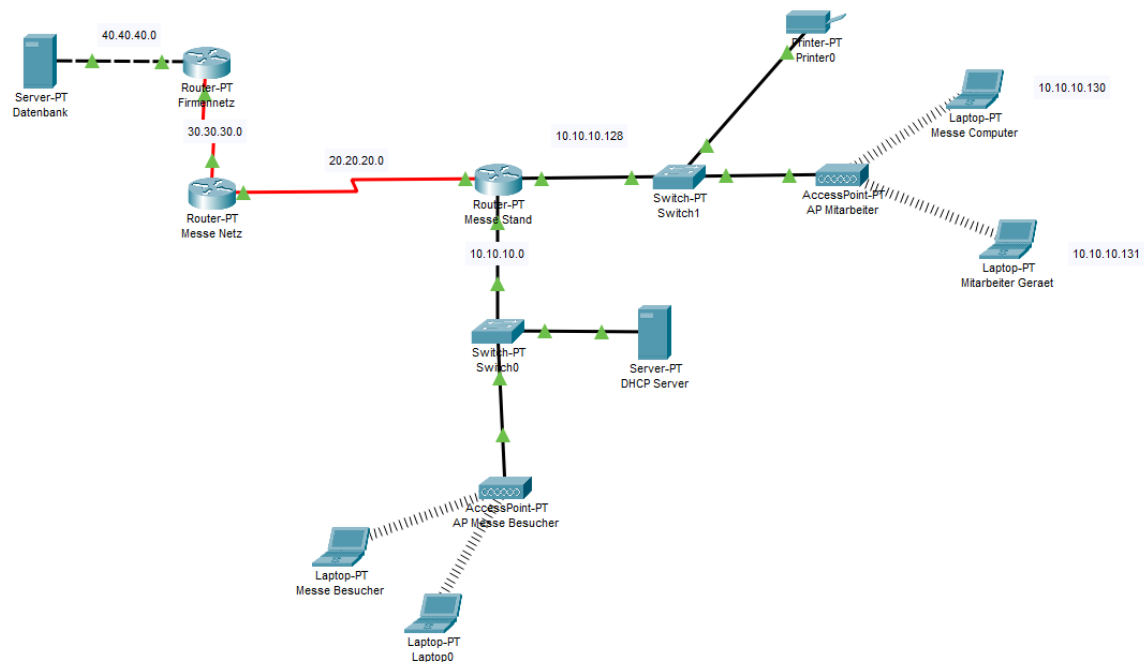
Bei dem Gästernetz steht jedem Gast das WLAN zur freien Verfügung. Dabei werden die IP-Adressen dynamisch verteilt. Im Gäste netz befindet sich auch das Gerät, mit dem sich die Besucher des Stands registrieren können.

Die Abbildung zeigt einen Entwurf der Netzwerkplanung.



3.1.2 Anbindung Messenetzwerk

Im Messenetz wird ein neuer Router hinzugefügt, welcher für den Messestand zur Verfügung steht. Das Netz für den Messestand wird in zwei Subnetze untergliedert mit den Netzmasken 10.10.10.0 und 10.10.10.128. So haben jeweils 126 Hosts im Gästernetz und 62 Hosts im Adminnetz Platz. Alle Hosts werden über Switches miteinander verbunden. So werden zwei Switches für den Messestand benötigt. Außerdem noch zwei Accesspoints, einen Drucker und einen DHCP Server. In der nachfolgenden Abbildung ist der DHCP-Server ein eigener Server. Am Messestand kann dies auch ein Modem übernehmen, der dazu in der Lage ist.



Vom Betreiber des Messestands müssen mindestens zwei Geräte bereitgestellt werden. Das eine Gerät dient für die Registrierung der Kunden und ist den Kunden auch bereitgestellt. Bei dem zweiten Gerät handelt es sich um einen Admin PC, welcher mit einem Passwort gesichert werden muss und nur von den Administratoren bedient werden soll.

[siehe Anhang „Messeplan“]

3.1.3 Netzwerk Einrichtung und IP-Zuweisung

Die Besucher bekommen ihre IP-Adressen dynamisch vom DHCP-Server im Gäste-Subnetz (10.10.10.0) zugewiesen während den Admins statische IP-Adresse im Admin-Netz (10.10.10.128) bekommen. Die IP-Adressen 10.10.10.130 und 10.10.10.131 werden zwei möglichen Admin PCs zugewiesen. In diesem Fall wird nur ein Admin-PC für die Anwendung benötigt.

Die Netze sind wie folgt aufgebaut:

	AP Mitarbeiter	AP Messe Besucher	Firmennetz
Netzwerkgröße	/26	/25	/28
Gateway	10.10.10.129	10.10.10.1	40.40.40.1
Netzmaske	255.255.255.192	255.255.255.128	255.255.255.0
Erste IP-Adresse	10.10.10.129	10.10.10.1	40.40.40.1
Letzte IP-Adresse	10.10.10.191	10.10.10.127	40.40.40.254
Broadcast	10.10.10.192	10.10.10.128	40.40.40.255
Anzahl Hosts	62	126	30

Das Firmennetz liegt außerhalb des Messenetzes und soll über einen VPN erreichbar sein.

3.1.4 Routing

Routing ist der Prozess, bei dem Datenpakete von einem Netzwerkgerät zu einem anderen übertragen werden, indem sie von einem Netzwerkgerät zum nächsten weitergeleitet werden. Dies geschieht, indem jedes Netzwerkgerät in einem Netzwerk eine Tabelle mit Routen hat, die festlegt, wohin Datenpakete gesendet werden sollen, basierend auf ihren Zieladressen.

Wenn ein Datenpaket von einem Gerät gesendet wird, prüft das Gerät zunächst, ob es das Zieladresse des Pakets ist. Wenn dies nicht der Fall ist, leitet das Gerät das Paket an das nächste Netzwerkgerät weiter, das auf der Grundlage der Routentabelle als nächstes in der Lage ist, das Paket dem Zieladresse näher zu bringen. Dieser Prozess wiederholt sich, bis das Paket sein Ziel erreicht hat.

Im Folgenden finden sich somit die statischen Routen zwischen dem Stand-, Messe- und Firmennetz.

Router	Ziel	Next Hop
Messe Stand	40.40.40.0/24	20.20.20.2
Messe Netz	10.10.10.128/25	20.20.20.1
Messe Netz	10.10.10.0/25	20.20.20.1
Messe Netz	40.40.40.0/24	30.30.30.2
Firmennetz	10.10.10.0/25	30.30.30.1
Firmennetz	10.10.10.128/25	30.30.30.1

Um weitere statische Routen (im Cisco Packet Tracer) hinzuzufügen muss wie folgt vorgegangen werden:

1. Öffnen Sie das Gerät, indem Sie darauf doppelklicken, und wählen Sie dann das "Routing" - Tab.
2. Klicken Sie auf das Symbol "Neue statische Route hinzufügen", um einen neuen Eintrag in der Routentabelle zu erstellen.
3. Geben Sie die Zieladresse und das Netzwerkmasken ein, für die die Route gilt.
4. Geben Sie den nächsten Hop ein, zu dem das Datenpaket gesendet werden soll, wenn es an das Gerät gesendet wird. Dies kann entweder eine IP-Adresse oder der Name eines anderen Geräts im Netzwerk sein.
5. Klicken Sie auf "OK", um die Route zu speichern.

3.2 WLAN

3.2.1 Zugang und Sicherheit

Die WLANs sind mit einem Passwort ausgestattet. Die Passwörter lauten:

- Für Admins
 - o SSID: AP_Mitarbeiter
 - o WPA-2: p4ssw0rd_1
- Für Gäste:
 - o SSID: AP_Messe_Besucher
 - o WPA-2: p4ssw0rd_2

Unter realen Umständen sollte mindestens für das Adminnetz ein sichereres Passwort gewählt werden. Für eine gesicherte Lernumgebung reicht ein einfaches Passwort, das zu Testzwecken sich leicht gemerkt werden kann.

3.2.2 Anbindung von Clients

Gäste dürfen sich bei den Standbetreibern das Passwort für das Gästernetz abholen und für eine begrenzte Zeit nutzen. Das Netz ist frei verfügbar und vom Adminnetz getrennt. Neue Gäste bekommen ihre IP dynamisch vom DHCP-Server vergeben.

Administratoren PCs bekommen eine feste IP-Adresse vergeben. Es ist nicht vorgesehen, dass mehr als zwei Admin PCs im Adminnetz enthalten sind. Jedoch ist es möglich mehr Geräte einzubinden. Dies ist optional und kann von den Standbetreibern selbst entschieden werden. Es können 126 Hosts im Adminnetz untergebracht werden.

3.3 Inbetriebnahme

Sobald der Stand zur Verfügung steht und die Geräte angeschlossen werden, müssen folgende Schritte getan werden:

Im Gästernetz:

1. Beim Gerät die dynamische IP-Adressen aktivieren, wenn dies nicht bereits getan ist.
2. Geräte mit dem Access Point verbinden und Passwort eingeben (Hier kann auch eine LAN-Verbindung genutzt werden).

Der DHCP Server wird von uns fertig konfiguriert ausgeliefert und muss nicht weiter konfiguriert werden. Dies gilt auch für die Access Points.

Im Adminnetz:

1. Drucker mit einem Kabel an den Switch anschließen.
2. Einem Gerät, das angeschlossen werden soll (bspw. PC) eine feste IP-Adresse zuweisen. Hier muss aus dem Bereich zwischen 10.10.10.130 und 10.10.10.191 gewählt werden. Keine IP-Adresse darf zwei Mal genutzt werden.
3. Gerät mit dem Access Point verbinden und Passwort für das WLAN eingeben (Hier kann auch eine LAN-Verbindung genutzt werden).

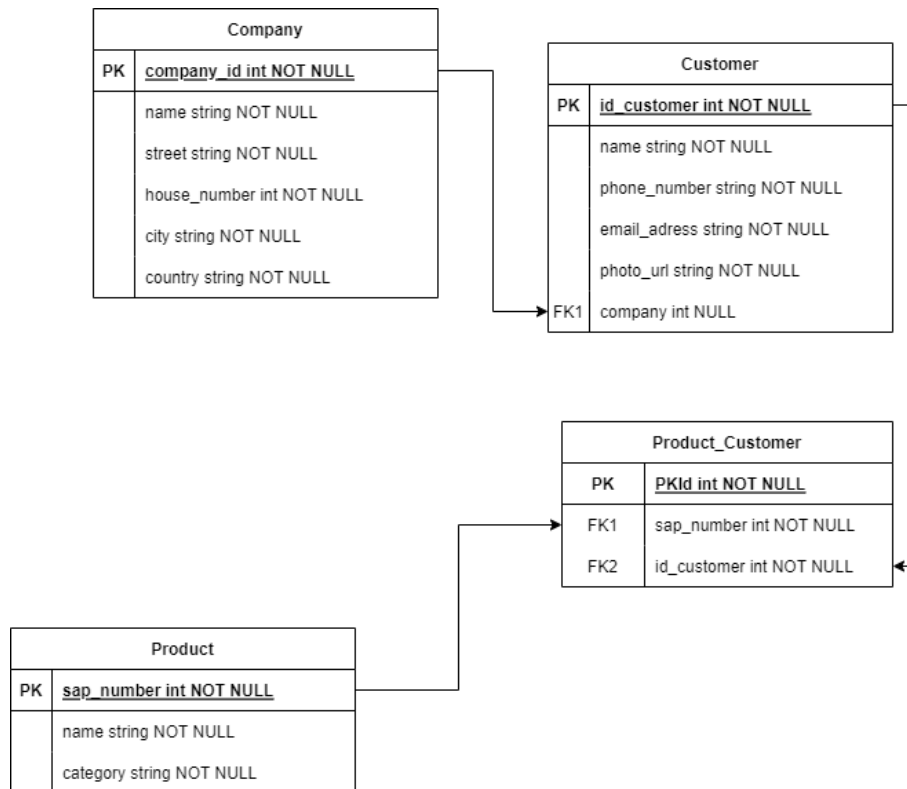
Der Router ist bereits für beide Netze konfiguriert und sofort betriebsbereit.

4. Teilprojekt SAE

4.1 Datenbank

4.1.1 Datenbankmodell z.B. Relationen-Modell

Bei der Datenbank werden vier Tabellen benötigt.



Datenspeicherung:

Die Daten werden bei der Registrierung zunächst offline gespeichert, so dass es zu jeder Zeit möglich ist, sich neu zu registrieren. Kunden können sich nicht erneut anmelden, da dies nicht benötigt wird. Bei einem Ausfall des Internets oder fehlender Verbindung zur Datenbank bleibt die Möglichkeit zur Registrierung bestehen.

Die Daten werden zunächst in einer JSON gespeichert. Vom Admin PC aus kann die JSON dann ausgelesen und in die Datenbank hochgeladen werden. In diesem Moment wird der Inhalt der JSON gelöscht. Es kann nicht zu einem Eintrags- und Löschkonflikt kommen, da die Daten des Kunden bei der Registrierung alle auf einmal in die JSON geschrieben werden. Werden die Daten hochgeladen, kann dies nicht zeitlich passieren.

Sobald die Daten hochgeladen werden, werden sie in einer MySQL Datenbank gespeichert. Für die Datenspeicherung ist ein Relationales Datenbankmodell vorgesehen.

Filter

Beinhalten das Wort:

Tabelle	Aktion	Datensätze	Typ	Kollation	Größe	Überhang
<input type="checkbox"/> company	★ Anzeigen Struktur Suche Einfügen Leeren Löschen	3	InnoDB	utf8mb4_general_ci	16,0 KiB	-
<input type="checkbox"/> customer	★ Anzeigen Struktur Suche Einfügen Leeren Löschen	2	InnoDB	utf8mb4_general_ci	32,0 KiB	-
<input type="checkbox"/> product	★ Anzeigen Struktur Suche Einfügen Leeren Löschen	4	InnoDB	utf8mb4_general_ci	16,0 KiB	-
<input type="checkbox"/> product_customer	★ Anzeigen Struktur Suche Einfügen Leeren Löschen	0	InnoDB	utf8mb4_general_ci	48,0 KiB	-
4 Tabellen	Gesamt	9	InnoDB	utf8mb4_general_ci	112,0 KiB	0 B

☐ Alle auswählen

markierte:

Drucken Strukturverzeichnis

Erzeuge Tabelle

Name:
Anzahl der Spalten:

OK

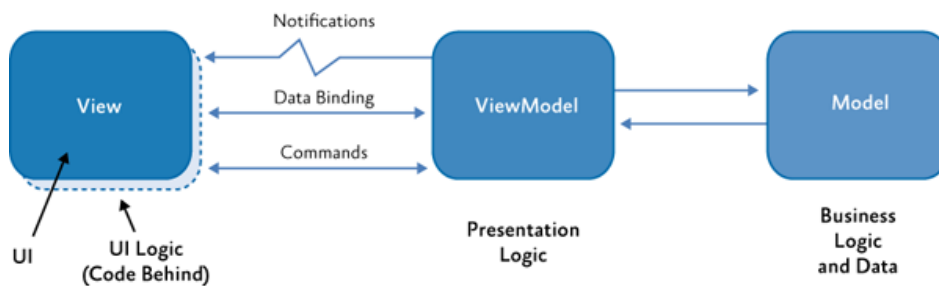
4.2 Aufbau und Funktionsweise

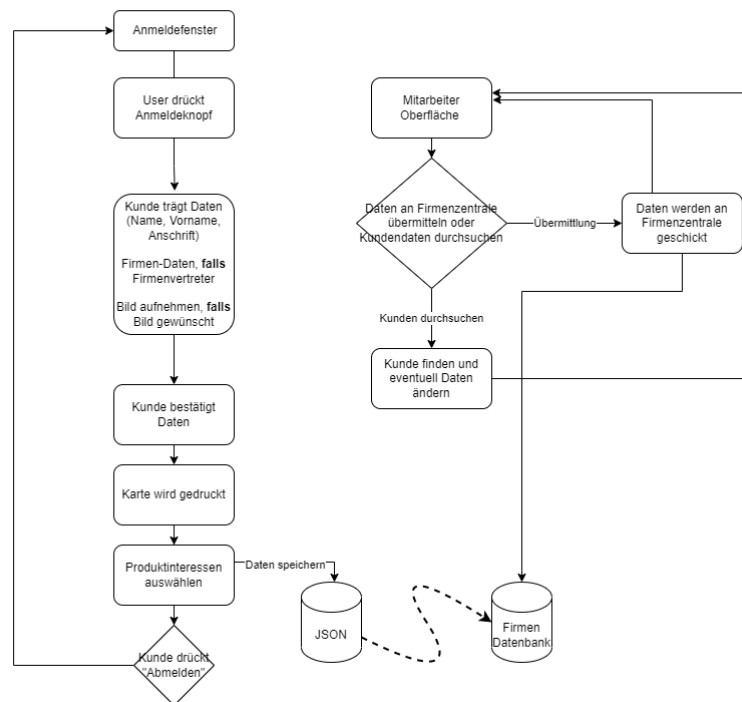
4.2.1 Architektur

Model View Controller:

MVVM (Model-View-ViewModel) C# ist eine Technik zur Erstellung von Client-Anwendungen, die die Kernfunktionen der WPF-Plattform steuern und das Testen der App-Funktionalität erleichtern. MVVM ist ein architektonisches Muster, das für drei verschiedene Komponenten steht. Es handelt sich um ein Entwurfsmuster für Software, das so gestaltet ist, dass die Bedienelemente der Benutzeroberfläche und die Logik des Programms voneinander getrennt sind. [1]

Das Model-View-ViewModel ist ein Software-Entwurfsmuster, das drei einzelne Komponenten symbolisiert, nämlich Model, View und ViewModel. [1]



Ablaufdiagramm:

Unterschieden wird zwischen dem Mitarbeiter und dem Kunden, der sich registrieren möchten. Beide haben eine eigene Oberfläche bzw. ein eigenes Programm, das auf einem eigenen PC läuft.

Kunde:

Sobald der Kunde sich registrieren möchte, kann er auf dem nächsten Fenster seine Daten eintragen und ein Bild aufnehmen (optional). Außerdem wird zwischen Firmenkunden und Nicht-Firmenkunden unterschieden. Firmenkunden geben dann auch noch ihre Firma ein. Am Ende wählt der Kunde noch seine Interessen im selben Fenster aus. Danach erfolgt die Datenspeicherung in der JSON. Sind alle Datenfelder ausgefüllt kann der Kunde diese durch den „Submit“-Button bestätigen und die Kundenkarte wird gedruckt.

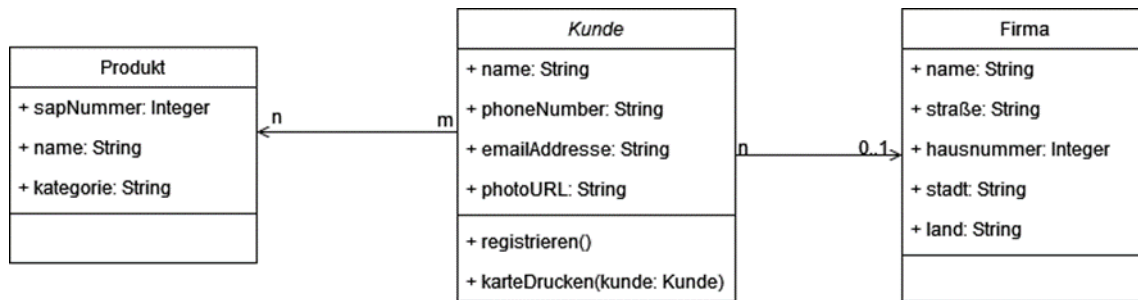
Danach wird ihm ein Dankesfenster angezeigt und der nächste Kunde kann sich registrieren.

Admin:

Administratoren melden sich auf einem anderen Gerät mit ihren Zugangsdaten an und können die Daten aus der JSON in die Datenbank speichern. Außerdem können sie sich die Interessen der einzelnen Kunden anzeigen lassen.

4.2.2 USE Case und UML Diagramme

UML Diagramm:

USE Cases:

A. Firmenkunde registriert sich.

Hauptakteur: Firmenkunde

Aufgabe: Ein Firmenkunde registriert sich im Portal.

1. Der Firmenkunde gibt seinen Vor- und Nachnamen ein.
2. Firmenkunde setzt einen Haken bei der Option „Firmenkunden“
3. Firmenkunde füllt seine Adresse aus.
4. Firmenkunde kann ein Bild aufnehmen.
5. Firmenkunde bestätigt seine eingegebenen Daten durch einen Klick auf den „Submit“-Button.
6. Kundenkarte für Firmenkunde wird gedruckt.

B. Kunde registriert sich.

Hauptakteur: Kunde

Aufgabe: Ein Kunde registriert sich im Portal.

1. Der Kunde gibt seinen Vor- und Nachnamen ein.
2. Kunde setzt keinen Haken bei der Option „Firmenkunden“.
3. Kunde kann ein Bild aufnehmen.
4. Kunde bestätigt seine eingegebenen Daten durch einen Klick auf den „Submit“-Button.
5. Kundenkarte für Kunde wird gedruckt.

C. Administrator speichert Daten.

Hauptakteur: Administrator

Aufgabe: Daten hochladen

1. Administrator drückt auf „Save JSON“
2. Daten werden aus der JSON in die Datenbank geschrieben.

D. Administrator ruft alle Daten ab.

Hauptakteur: Administrator

Aufgabe: Daten ansehen

1. Administrator drückt auf einen Kunden in der Liste
2. Alle Daten über die Kunden mit ihren Interessen werden in einer Tabelle angezeigt.

E. Produktliste aktualisieren

Hauptakteur: Administrator

Aufgabe: Lokale Produkte aktualisieren

1. Administrator öffnet die Mitarbeiter Applikation.
2. Administrator trägt neue Produkte ein, löscht alte oder aktualisiert bestehende.
3. Neue Produkte werden aus der Datenbank in einer JSON gespeichert.

4.2.3 Prerequisites: Bibliotheken und Komponenten

- .NET (6) Framework

.NET ist ein Softwareframework, das von Microsoft entwickelt wurde und auf dem Windows-Betriebssystem ausgeführt wird. .NET bietet eine Laufzeitumgebung und eine umfassende Bibliothek von Klassenbibliotheken, die von Entwicklern verwendet werden können, um Anwendungen für Windows, Web und mobile Plattformen zu entwickeln.

Es wird im Projekt verwendet, um die Entwicklung zu vereinfachen. So können ausführbare Anwendungen leichter erstellt werden.

- WPF

WPF steht für Windows Presentation Foundation und ist ein Teil von Microsoft .NET, der es Entwicklern ermöglicht, Benutzeroberflächen für Windows-Anwendungen zu erstellen. WPF verwendet XML-basierte XAML-Markup-Sprache zum Definieren von Benutzeroberflächen, die auf der .NET Framework-Laufzeit ausgeführt werden.

- Newtonsoft Json

Newtonsoft Json ist eine Bibliothek, die es ermöglicht, JSON-Daten zu serialisieren und deserialisieren. Man kann auch benutzerdefinierte Serialisierungsregeln definieren und die JSON-Serialisierung und -Deserialisierung anpassen, um den spezifischen Anforderungen zu entsprechen.

- MySQL.Data

Die Bibliothek ermöglicht es, mit einer MySQL-Datenbank zu interagieren. Sie bietet eine Implementierung des ADO.NET-Providers für MySQL, der es Entwicklern ermöglicht, in C# mit einer MySQL-Datenbank zu arbeiten, indem er eine Schnittstelle bereitstellt, die dem ADO.NET-Modell entspricht.

Mit der MySQL.Data Bibliothek kann man die Verbindung zur Datenbank herstellen, Befehle ausführen, um Daten abzufragen oder zu ändern und Transaktionen verwalten.

- Aforge

Die Bibliothek umfasst eine Reihe von Algorithmen für die Muster-Erkennung, die Bild-Stabilisierung, die Farbanalyse, die Verfolgung von Objekten, die Identifizierung von Barcodes und QR-Codes, die OCR und die Augmented Reality. Mit ihr kann man...

- Bilder und Videos lesen, aufnehmen und speichern
- Bilder und Videos filtern, verarbeiten und transformieren (z.B. Skalieren, Rotieren, Spiegeln)
- Bilder und Videos segmentieren und analysieren (z.B. Kanten erkennen, Merkmale extrahieren, Gesichter erkennen)
- Bilder und Videos synthetisieren und generieren (z.B. synthetische Bilder erstellen, Text auf Bilder schreiben)
- Bilder und Videos komprimieren und dekomprimieren (z.B. JPEG, GIF, PNG)

4.2.4 Inbetriebnahme vor Ort

Das Programm ist eine portable .exe-Datei, die ohne Installation auf den entsprechenden Rechner laufen kann. Dabei läuft auf dem Kunden-PC die Software für die Kunden und auf dem Admin-PC die Verwaltungssoftware.

Auf dem Kunden-PC muss ein Ordner im „Downloads“-Ordner angelegt werden und dieser muss für den Admin-PC freigegeben werden. In diesem Ordner wird die JSON angelegt, in der die Kundendaten gespeichert werden und auf die die Admins zugreifen können, um diese hochzuladen.

Die Applikation auf dem Kunden-PC läuft im gesicherten Vollbildmodus, so dass Kunden nicht auf die JSON zugreifen können und in der App „gefangen“ sind.

4.2.5 Technische Beschreibung der WebCam-Anbindung

Um auf die Kamera zuzugreifen nutzen wir die Aforge Bibliothek. Mit dieser kann man, wie bereits in Prerequisites erwähnt, Bilder und Videos aufnehmen, bearbeiten und analysieren. Ein einfaches Beispiel, um mit der Bibliothek auf die Standardkamera zuzugreifen ist wie folgt.

```
1  using AForge.Video;
2  using AForge.Video.DirectShow;
3  using System.Drawing;
4
5  // ...
6
7  FilterInfoCollection videoDevices;
8  VideoCaptureDevice videoSource;
9
10 // Enumerate video devices
11 videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
12
13 // Check if at least one video device was found
14 if (videoDevices.Count == 0)
15     throw new ApplicationException();
16
17 // Create video source
18 videoSource = new VideoCaptureDevice(videoDevices[0].MonikerString);
19
20 // Set NewFrame event handler
21 videoSource.NewFrame += new NewFrameEventHandler(video_NewFrame);
22
23 // Start the video source
24 videoSource.Start();
25
26 // ...
27
28 private void video_NewFrame(object sender, NewFrameEventArgs eventArgs)
29 {
30     // get new frame
31     Bitmap bitmap = eventArgs.Frame;
32
33     // process the frame
34     // ...
35
36     //save the Image
37     bitmap.Save("image.jpg", ImageFormat.Jpeg);
38 }
```

Dabei wird die VideoCaptureDevice-Klasse verwendet, um eine Verbindung zu einer Webcam oder einer anderen Videoquelle herzustellen, und dann das NewFrame-Ereignis behandelt, um auf ein neues Bild zu reagieren. Sobald man ein neues Bild hat, kann man es mit der Save()-Methode einer Bitmap-Klasse speichern.

In unserem konkreten Fall brauchen wir drei Klassen.

```

184 private void StartCamera(object sender, RoutedEventArgs e)
185 {
186     try
187     {
188         infoCollection = new FilterInfoCollection(FilterCategory.VideoInputDevice);
189         videoCaptureDevice = new VideoCaptureDevice(infoCollection[0].MonikerString);
190         videoCaptureDevice.NewFrame += FinaleFrame_newFrame;
191         videoCaptureDevice.Start();
192         SaveButton.Visibility = Visibility.Visible;
193         ImagePlaceholder.Visibility = Visibility.Collapsed;
194     }
195     catch (ArgumentOutOfRangeException ex)
196     {
197         MessageBoxHelper.ThrowErrorMessageBox(
198             "An error occurred while starting your camera. \nDo you have a camera installed?",
199             "Start camera error");
200     }
201     catch (Exception ex)
202     {
203         MessageBoxHelper.ThrowErrorMessageBox(
204             "An error occurred while starting your camera. Please try again. \n" + ex.Message,
205             "Start camera error");
206     }
207 }

```

Die Methode "StartCamera":

- Erstellt eine neue FilterInfoCollection und speichert es in der Variablen "infoCollection", diese sammelt Informationen über verfügbare Video Eingabe Geräte (Webcams)
- Erstellt ein neues VideoCaptureDevice mit dem MonikerString des ersten gefundenen Geräts in der "infoCollection"-Variable
- fügt dem VideoCaptureDevice das Ereignis "NewFrame" hinzu, welches die Methode "FinaleFrame_newFrame" aufruft
- startet das VideoCaptureDevice
- Setzt die Sichtbarkeit des "SaveButton" auf sichtbar und die des "ImagePlaceholder" auf unsichtbar
- Fängt mögliche Ausnahmen ab, die auftreten können, wenn es keine Kamera vorhanden ist oder ein anderer Fehler auftritt und zeigt dem Benutzer eine Fehlermeldung an

```

208 private void SaveImage(object sender, RoutedEventArgs e)
209 {
210     try
211     {
212         videoCaptureDevice.NewFrame -= FinaleFrame_newFrame;
213         Image img = System.Drawing.Image.FromStream(memoryStream);
214         videoCaptureDevice.SignalToStop();
215
216         img.Save(downloadDirectory + "\\image.jpeg", ImageFormat.Jpeg);
217         Webcam.Source = new BitmapImage(new Uri(downloadDirectory + "\\image.jpeg"));
218         img.Dispose();
219         SaveButton.Visibility = Visibility.Collapsed;
220     }
221     catch (Exception ex)
222     {
223         MessageBoxHelper.ThrowErrorMessageBox(
224             ex.Message + "\nAn error occurred while capturing your image. Please take one again.",
225             "saving error");
226     }
227     finally
228     {
229         memoryStream.Dispose();
230     }
231 }

```

Die Methode "SaveImage":

- Entfernt das "NewFrame"-Ereignis von dem VideoCaptureDevice
- erstellt ein neues Image-Objekt aus dem Inhalt des "memoryStream"
- beendet das VideoCaptureDevice
- speichert das Image-Objekt als JPEG-Datei in einem bestimmten Verzeichnis mit dem Namen "image.jpeg"
- zeigt das gespeicherte Bild in dem Webcam-Objekt an
- gibt die Ressourcen des Image-Objekts und des memoryStream frei
- Setzt die Sichtbarkeit des "SaveButton" auf unsichtbar
- Fängt mögliche Ausnahmen ab, die beim Speichern des Bildes auftreten können und zeigt dem Benutzer eine Fehlermeldung an.

```

116 private void FinaleFrame_newFrame(object sender, NewFrameEventArgs eventArgs)
117 {
118     try
119     {
120         using (var bitmap = (Bitmap)eventArgs.Frame.Clone())
121         {
122             bitmapImage = new BitmapImage();
123             bitmapImage.BeginInit();
124             memoryStream = new MemoryStream();
125             bitmap.Save(memoryStream, ImageFormat.Bmp);
126             bitmapImage.StreamSource = memoryStream;
127             bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
128             bitmapImage.EndInit();
129         }
130         bitmapImage.Freeze();
131         Dispatcher.BeginInvoke(new ThreadStart(delegate
132             { Webcam.Source = bitmapImage; }));
133     }
134     catch (Exception ex)
135     {
136         //catch your error here
137     }
138 }
139
140

```

In der Methode:

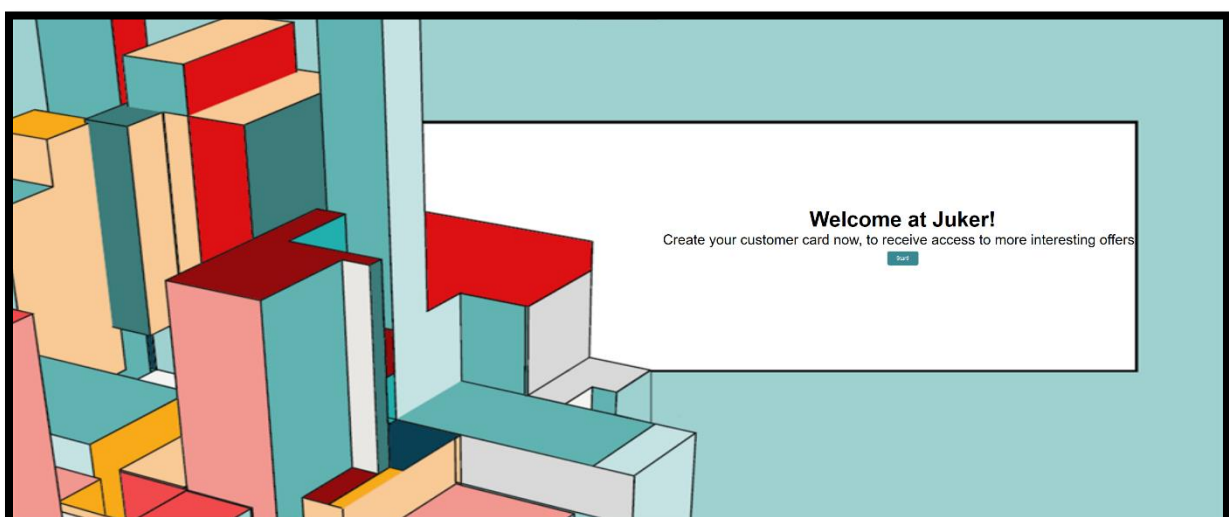
- wird eine temporäre Bitmap Variable erstellt, die das aktuelle Frame des Videos enthält.
- wird eine neue BitmapImage-Variable erstellt, die aus dem Frame des Videos erstellt wird
- wird eine neue MemoryStream-Variable erstellt
- das aktuelle Frame wird in die memorystream gespeichert, mit dem Format BMP
- die StreamSource-Eigenschaft der BitmapImage-Variable wird auf die memorystream festgelegt
- wird die Eigenschaft "CacheOption" auf "OnLoad" gesetzt
- wird die BitmapImage-Variable eingefroren und in die Dispatcher-Queue gestellt
- wird das Quellbild des Webcam-Objekts auf die BitmapImage-Variable festgelegt

Es wird auch ein try-catch-Block verwendet, um Errors abzufangen, die bei der Ausführung entstehen können.

Zusammengefasst wird jedes empfangene Frame verarbeitet, indem es in eine Bitmap gespeichert wird, diese in eine BitmapImage konvertiert und in einer MemoryStream gespeichert wird, damit es später verwendet werden kann. Dann wird das Bild an die GUI übergeben, damit es angezeigt werden kann.

4.2.6 Anleitung Bedienung durch den Kunden

1. Beim Start sieht man einen Willkommensbildschirm, über den der Kunde zum Anmeldeformular kommt



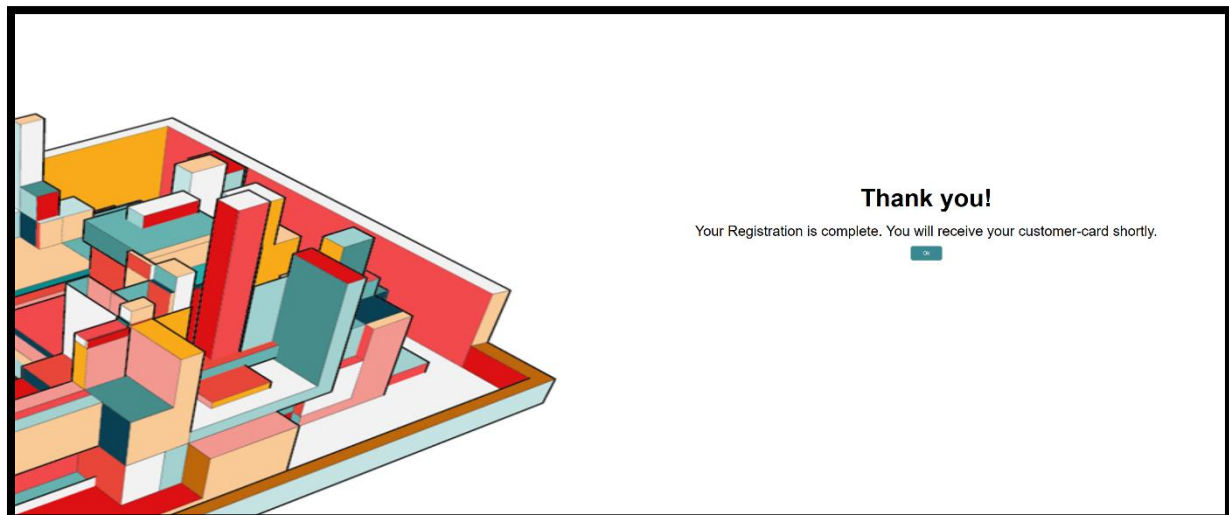
2. Das Formular muss mit persönlichen Daten ausgefüllt werden und es gibt die Möglichkeit ein Bild aufzunehmen. Dann sollte man seine Daten auch kontrollieren. Danach muss auf „Submit“ gedrückt werden.

The screenshot shows a web registration form with the following elements:

- Registration** (Title)
- Image Upload:** A camera icon and a text prompt "Click the symbol to take a picture!" below a placeholder box.
- Form Fields:**
 - First Name (text input)
 - Last Name (text input)
 - Phone Number (text input)
 - Email (text input)
 - Company Customer (checkbox)
- Buttons:** "Submit" and "Cancel" (both green).
- Our Products:** A section with a table listing products and a 3D bar chart below it.

Id	Name	Category
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin
0	Bohrer	Maschin

3. Sobald auf den „Submit“-Button gedrückt wird, wird man zum Dankesfenster weitergeleitet und der Registrierungsprozess ist für den Kunden abgeschlossen. Die Kundenkarte wird gedruckt und kurz darauf erscheint das „Danke“-Fenster.



4. Danach drückt man auf „OK“ und der nächste Kunde kann sich registrieren.

4.2.7 Anleitung Datenabruf und Übermittlung

1. Um die Daten hochzuladen, drückt der Administrator auf „Save JSON“ und die Daten werden in der Datenbank gespeichert.

Customers	
First Name	Last Name
Man	Fred
Jürgen	Jürgensen

Product Interests	
Name	Category
product 1	Pleasure
product 2	project building
product 3	Service
product 4	Support

2. Um die Daten anzeigen zu lassen, drückt der Administrator auf irgendeinen Kunden aus der Liste. Rechts in der Anzeige werden dann die Interessen des Kunden aufgelistet.

Customers	
First Name	Last Name
Man	Fred
Jürgen	Jürgensen

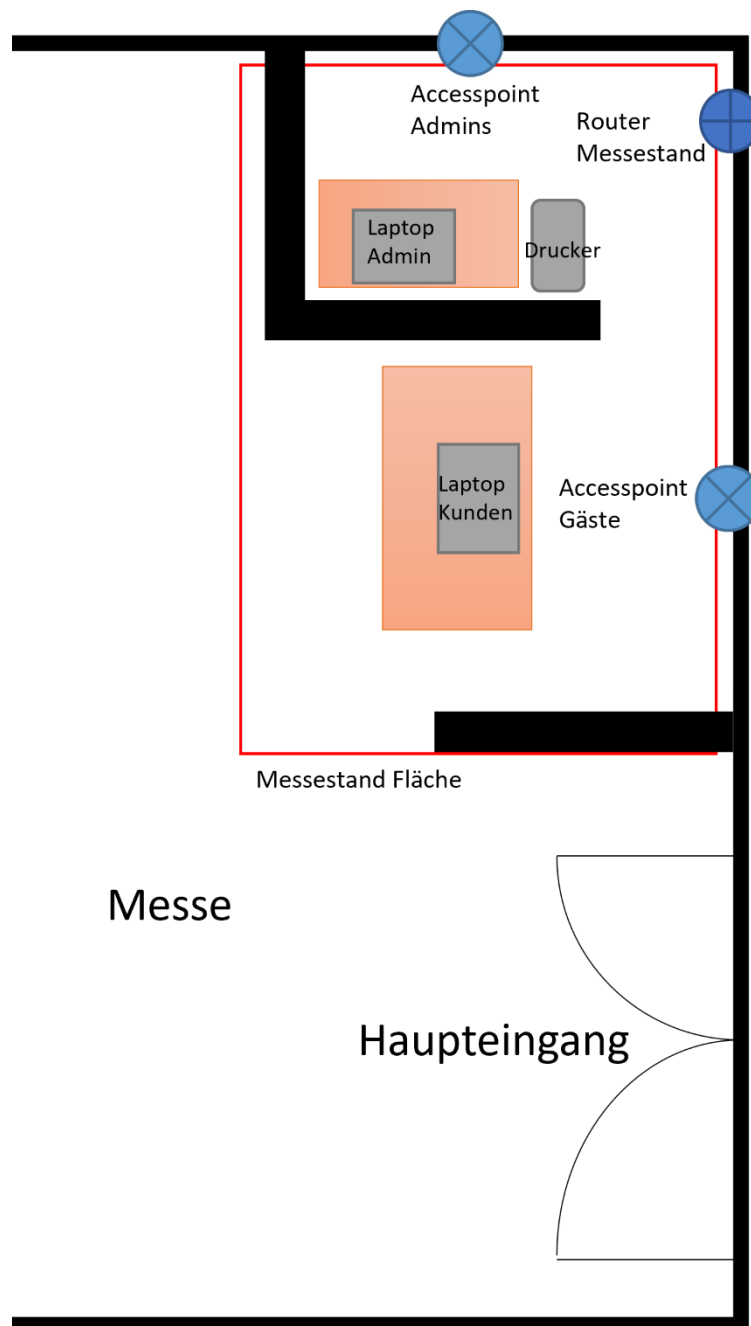
Product Interests	
Name	Category

4.2.8 Testszenarien

Test Nr.	Beschreibung	Erwartetes Ergebnis	Status
1	Kunde registriert sich und möchte ein Bild aufnehmen. Kundenkarte soll am Ende gedruckt werden.	Daten werden in der Datenbank gespeichert. Bild wird erfolgreich aufgenommen. Bestätigt wird dem Kunden angezeigt und Karte wird gedruckt	Die Daten werden erfolgreich in die JSON geschrieben und der Kunde wird zur nächsten Ansicht geführt.
2	Administrator möchte Daten aus der JSON in die Datenbank hochladen.	Daten werden aus der JSON in die Datenbank gespeichert. Daten sind in der Datenbank zu finden. Inhalt der JSON wird geleert, so dass keine Daten doppelt hochgeladen werden.	Daten werden erfolgreich aus der JSON in die Datenbank geschrieben und die JSON wird geleert.
3	Administrator ruft alle Daten aus der Datenbank ab.	Daten werden dem Admin in einer Tabelle angezeigt.	Daten können erfolgreich abgerufen werden.
4	Kunde nimmt ein Bild von sich auf.	Ein Kunde soll mit der zur Verfügung gestellten Laptop Kamera ein Bild aufnehmen, dass in der Datenbank gespeichert wird.	Bild wird aufgenommen und erfolgreich gespeichert.

Anhang

Messeplan



Verweise

- [1] Educba, „Educba,“ [Online]. Available: <https://www.educba.com/mvvm-c-sharp/>. [Zugriff am 3 Januar 2023].
- [2] „Prism Library,“ [Online]. Available: <https://prismlibrary.com/docs/wpf/legacy/Implementing-MVVM.html>. [Zugriff am 3 Januar 2023].