**QR Zutrittsscanner**

Anforderungsdokumentation

Autoren: Mitja Perko, Urs Heusser

Chur, 23.01.2023

**Inhaltsverzeichnis**

[1 Management Summary 3](#_Toc121056248)

[1.1 Auftraggeber 3](#_Toc121056249)

[1.2 Problemstellung 3](#_Toc121056250)

[1.3 Zusammenfassung 3](#_Toc121056251)

[2 Einleitung, Ziel 4](#_Toc121056252)

[2.1 Einleitung 4](#_Toc121056253)

[2.2 Ziel 4](#_Toc121056254)

[3 Ausgangslage 5](#_Toc121056255)

[3.1 Problemstellung 5](#_Toc121056256)

[3.2 Lösung 5](#_Toc121056257)

[4 Hauptteil 6](#_Toc121056258)

[4.1 Planung 6](#_Toc121056259)

[4.1.1 Priorisiertes Backlog 6](#_Toc121056260)

[4.2 Architektur 6](#_Toc121056261)

[4.2.1 System Skizze 6](#_Toc121056262)

[4.3 Server 7](#_Toc121056263)

[4.4 Client 7](#_Toc121056264)

[4.4.1 Mockup 7](#_Toc121056265)

[4.5 Continuous Integration 8](#_Toc121056266)

[5 Abbildungsverzeichnis 9](#_Toc121056267)

[6 Anhänge 10](#_Toc121056268)

# Management Summary

## Auftraggeber

Für diese Arbeit gibt es keinen Auftraggeber.

Die Aufgabe ist jedoch auf eine Anfrage aus unserer Firma zurück zu führen und somit ein „real life“ Beispiel.

## Problemstellung

Eine Firma im Raum Chur möchte den Mitarbeitenden freien Eintritt ins Hallenbad Obere Au ermöglichen.

Das QR Scanner System soll es ermöglichen, die Mitarbeitenden zu erkennen und der Firma Informationen zur Nutzung des Angebots zu geben.

Die Mitarbeitenden werden im System erfasst und eine Quittung mit persönlichem QR-Code wird den Mitarbeitenden ausgestellt.

Das Kassenpersonal im Hallenbad scannt den QR-Code, eine Identifizierung inklusive Status wird ermöglicht.

Die Firma erhält vom Hallenbad eine Rechnung über die genutzten Eintritte.

## Zusammenfassung

Das QR Scanner System digitalisiert einen Prozess auf einfache Art und Weise.

Das System ist schlank gehalten, damit der Prozess einfach zu nutzen ist und dabei einen möglichst hohen Nutzen bringt.

Unser Fokus in dieser Semesterarbeit liegt darauf das im Modul Continuous Integration gelernte anzuwenden.

Mit gitHub Actions wollen wir unsere Softwareentwicklungs-Workflows automatisieren. Die Unit Tests sollen bei jedem Push automatisch ausgeführt werden.

Das GUI wollen wir mit Selenium testen und versuchen auch dies automatisiert in gitHub Actions zu realisieren.

# Einleitung, Ziel

## Einleitung

## Ziel

# Ausgangslage

## Problemstellung

## Lösung

# Hauptteil

## Planung

### Priorisiertes Backlog

|  |  |  |
| --- | --- | --- |
| *ID* | *Titel* | *Priorität* |
| 1 | Allgemeine Bedienung | Hoch |
| 2 | Aufbau Oberfläche |  |
| 3 | Verwaltung: Liste 1 |  |
| 4 | Verwaltung: Liste 2 |  |
| 5 | Verwaltung: Neuer Eintrag |  |
| 6 | Verwaltung: Eintrag ändern |  |
| 7 | Zutrittsverwaltung: Kamera |  |
| 8 | Zutrittsverwaltung: QR-Code |  |
| 9 | Zutrittsverwaltung: Mitarbeitendendaten werden angezeigt |  |
| 10 | Zutrittsverwaltung: Erneutes Scannen |  |

## Architektur

### System Skizze

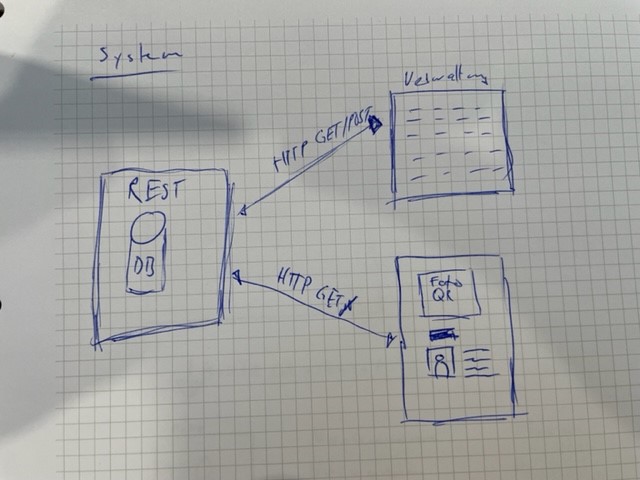
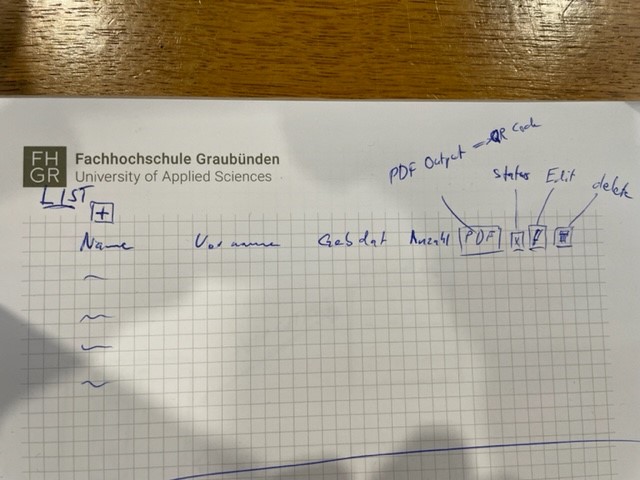


Abbildung 2: Systemskizze

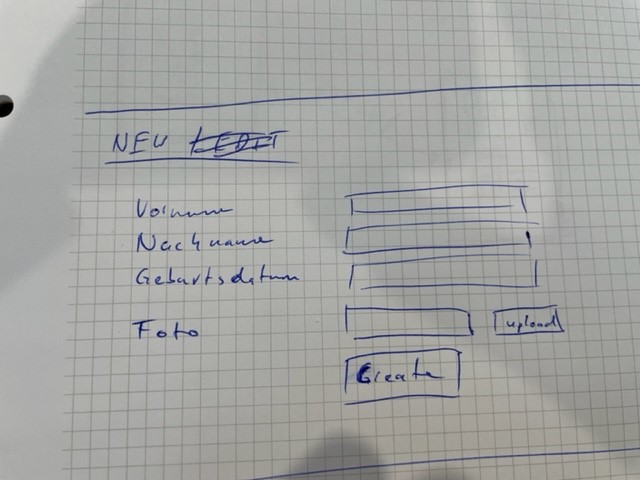
## Server

## Client

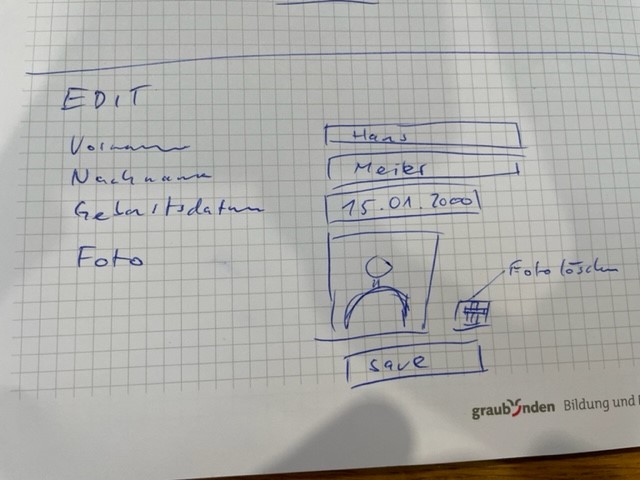
### Mockup



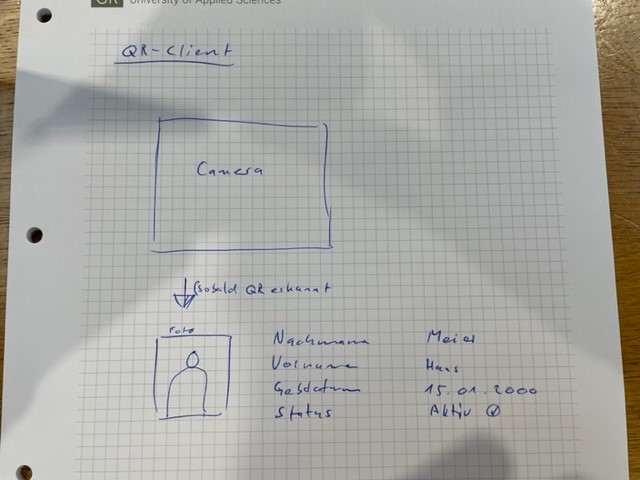
Die Administrationsoberfläche besteht im Kern aus einer Liste mit allen erfassten Mitarbeitenden. Hier kann die PDF Quittung mit dem QR-Code heruntergeladen, die Anzahl der Badeintritte angesehen und der Status des Mitarbeitenden geändert, neue Mitarbeitende erfasst oder bestehende verändert werden. Es ist gar möglich Mitarbeitendeneinträge zu löschen.



Die Erfassungsmaske für neue Mitarbeitende enthält je ein Eingabefeld für Vor-, Nachname, und das Geburtsdatum, sowie ein Upload-Feld für ein Foto.



Beim Ändern können sämtliche Daten der Mitarbeitenden angepasst werden. Das Foto kann gelöscht und neu hochgeladen werden.



Der QR-Client enthält ein Kamerabild, mit welchem der QR-Code gescannt werden kann. Sobald ein QR-Code erkannt wird, werden die gefundenen Mitarbeitendendaten ausgegeben und der Eintrittzähler auf dem Server um eins erhöht.

## Continuous Integration

# Abbildungsverzeichnis

[Abbildung 1: Use Case Diagramm 4](#_Toc104233181)

[Abbildung 2: Systemskizze 4](#_Toc104233182)

# Anhänge

Priorisiertes Backlog mit funktionalen- und nichtfunktionalen User Stories