

Dokumentation zu Anwesenheitsliste

3186523
9008480
6196929

Architekturdokumentation.....	2
Anwenderdokumentation	7
Dokumentation des Betriebs	9
Mitgliederbeitrag.....	10
Beitrag von 3186523	10
Beitrag von 9008480	10
Beitrag von 6196929	10

Architekturdokumentation

Aufbau

Die Anwendung besteht aus folgenden Bestandteilen:

- Konfigurationsdatei
- Webserver für die QR-Code Seite
- Webserver für die An- & Abmeldeseite
- Ordner für Journaldateien
- Analyse-Tool

Konfigurationsdatei

Bei der Konfigurationsdatei handelt es sich um die location.xml. Diese beinhaltet alle Orte, die besucht werden können.

Jeder Ort verfügt über eine ID, einen Namen, einen Accesstoken, einen Currenttoken und einen Oldtoken.

Der Accesstoken wird zur Generierung der URL und des QR-Codes genutzt. Er ist stationär und ändert sich nicht. Er beschreibt lediglich den zugehörigen Ort anhand einer Abkürzung.

Der Currenttoken bzw. Validitätstoken ist eine automatisch generierte Zahlenfolge, die dem Nutzer den Zugang zur Anmeldeseite ermöglicht. Der Currenttoken ist nur für eine bestimmte Zeit gültig. Die Dauer seiner Gültigkeit kann mit Hilfe einer Flag gesetzt werden. Er wird ebenfalls für die Generierung des QR-Codes benötigt.

Der Oldtoken speichert den vorherigen Currenttoken. Dieser wird benötigt, um dem Nutzer die Abmeldung auch mit dem vorherigen Token zu ermöglichen, damit kein neuer angefordert werden muss.

```

1  <Locations>
2  <location>
3      <id>1</id>
4      <name>Mosbach</name>
5      <accesstoken>rg</accesstoken>
6      <CurrentToken>925217946067037477</CurrentToken>
7      <OldToken>1367073459503061451</OldToken>
8  </location>
9  <location>
10     <id>2</id>
11     <name>Bad Mergendheim</name>
12     <accesstoken>erg</accesstoken>
13     <CurrentToken>14070193543771878181</CurrentToken>
14     <OldToken>14512049057207902155</OldToken>
15 </location>
16 </Locations>
17

```

Aufbau der location.xml

Eingelesen wird die Konfigurationsdatei in Locations.go.

Webserver für die QR-Code Seite

Der Webserver wird beim Ausführen der Anwendung gestartet. Er rendert zu Beginn das Template qr.html. Dieses beinhaltet ein Dropdown Menü, welches die aus der XML eingelesenen Orte als Auswahl anbietet. Übertragen wird die Auswahl dann mit der Post-Methode. Nachdem Absenden des Formulars findet ein Redirect statt und das qrSingle.html Template wird gerendert. Die URL wird beim Redirect so angepasst, dass der Accesstoken des ausgewählten Ortes an die URL angehängt wird. Die QR Single Seite sorgt dafür, dass der QR-Code erzeugt wird. Und das QR Single Template rendert die generierte .png-Datei des QR-Codes und stellt sie anschließend dar.

War der Redirect nicht erfolgreich oder die Seite wird mit einem falschen bzw. Nicht mehr aktuellen, also abgelaufenen Token aufgerufen, wird der Nutzer zurück auf die Ortsauswahl Seite zurückgeleitet.

Die URL, welche der QR-Code beinhaltet setzt sich aus dem Accesstoken und dem Currenttoken zusammen.

Der Aufbau ist wie folgt: `https://DOMAIN:PORT/?location=ACESSTOKEN&access=CURRENTTOKEN`

Da der QR-Code nur für bestimmte Zeit valide sein soll, sorgt das QR Single Template dafür, dass der QR-Code nach Ablauf der Validationtime neu erzeugt wird. Die Validationtime wird über die Flags gesetzt und für den QR-Code zusätzlich noch mit 1000 multipliziert, damit die Zeitangabe in Sekunden vorliegt.

Webserver für die An- & Abmeldeseite

Für die Anmeldung benötigt man den im vorherigen Abschnitt generierten QR-Code. Die im QR-Code beinhaltete URL ist die URL für die Anmeldeseite des entsprechenden Ortes. Die An- bzw. Abmeldeseite wird durch das login.html Template generiert. Das Template beinhaltet ein Formular mit den Input Feldern Name und Adresse. Übermittelt werden die Daten mit der Post-Methode. Die erhaltenen Anmeldedaten werden dann mit dem entsprechenden Ort und dem Status, also ob es eine An- oder Abmeldung war, in die entsprechende Journal-Datei geschrieben. Die Anmeldedaten werden auf dem Server gespeichert, damit bei erneuter Anmeldung oder bei der Abmeldung, mit demselben Token, der Nutzer nicht erneut seine Daten eingeben muss. Die Felder werden dann automatisch vorher ausgefüllt.

Ordner für Journaldateien

Im Ordner Journal wird für jeden Tag eine neue Journal-Datei angelegt. Bei der Journal-Datei handelt es sich um eine Textdatei. Die Namensgebung dafür sieht wie folgt aus: MM-DD-YYYY.txt

Eine Journal-Datei hält jede An- & Abmeldung fest, die am entsprechenden Tag erfolgt ist. Der Aufbau der Datei ist: Ort;Adresse;Name;Status;Zeit

Ort gibt hierbei an, an welchem Standort der Nutzer sich an- bzw. Abgemeldet hat. Adresse und Name sind die Daten, welche der Nutzer bei der Anmeldung angegeben hat. Und Status hält fest ob es sich um eine An- oder Abmeldung gehandelt hat. Entsprechend wird "Angemeldet" oder "Abgemeldet" eingetragen. Zeit hält die Uhrzeit fest zu der die An- bzw. Abmeldung erfolgt ist, nach dem Schema HH:MM. Pro Zeile wird nur ein Datensatz festgehalten.

Analyse-Tool

Das Analyse-Tool ist als Kommandozeilenprogramm angelegt und soll die Journaldateien auswerten. Folgende Funktionalitäten wurden hierfür implementiert:

- Einlesen einer Journaldatei
- Standortzurückverfolgung einzelner Personen
- Export einer Anwesenheitsliste für einen angegebenen Standort (Kompatibel mit MS Excel und LibreOffice)

Grundsätzlich benötigt jede Funktion die Angabe des Tags der mit MM-DD-YYYY codiert ist. Für die meisten Funktionen werden aber zusätzliche Parameter gebraucht wie z.B. den Namen des Studenten oder den Standort.

Für die das Einlesen eines Journals wird der Tag benötigt. Bei dem Einlesen wird der Inhalt in ein zweidimensionales Array gespeichert, das dem Schema Zeilen und Spalten folgt. Die Spalten pro Zeile lauten Standort, Adresse, Name, Status und Zeit. Die Einlesefunktion wird in den anderen Funktionen des Analyse-Tools benutzt.

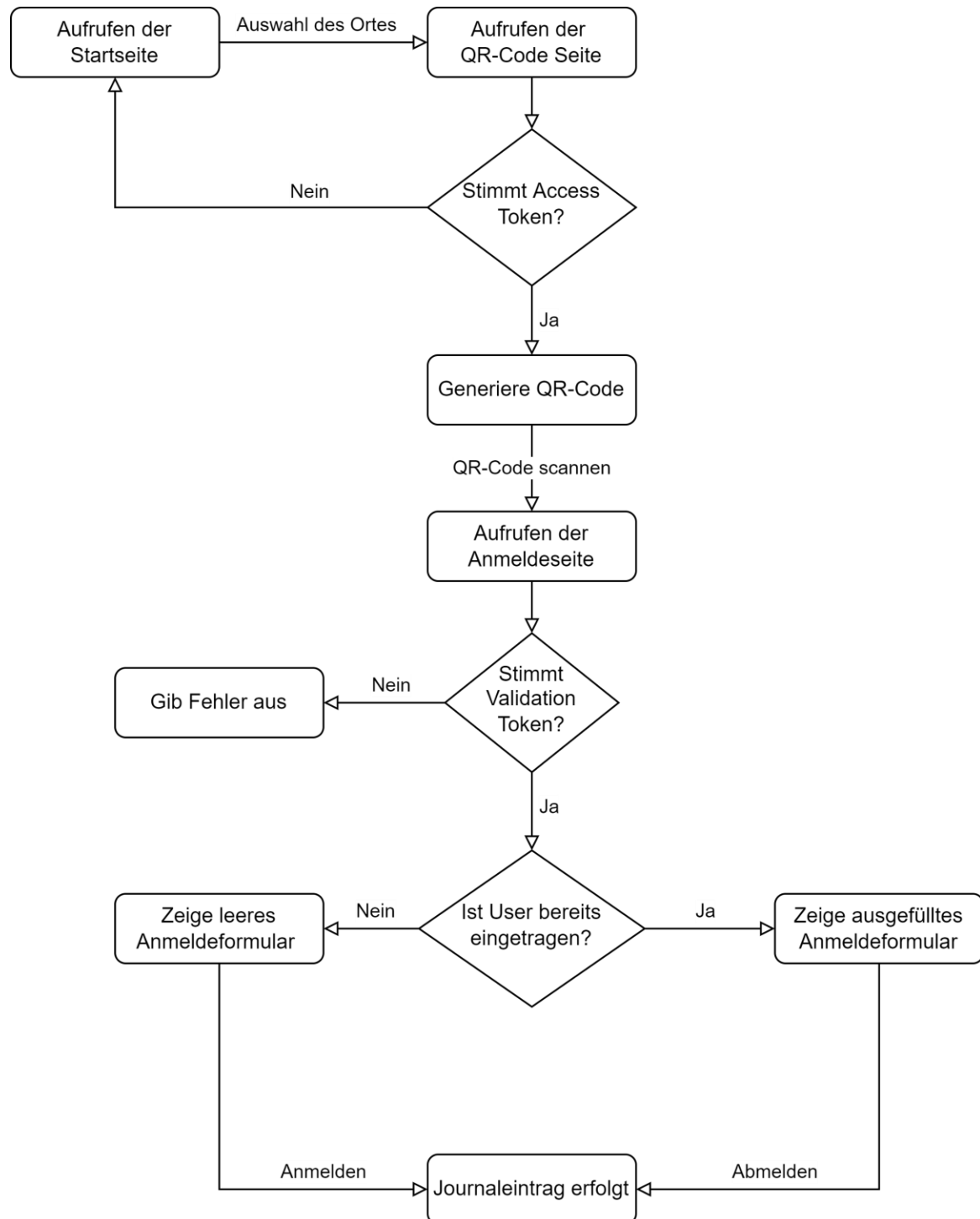
Für die Standortzurückverfolgung wird zusätzlich noch der Name gebraucht, der gesucht werden soll. Zu Beginn der Funktion wird das Journal des angegebenen Tages eingelesen. Anschließend werden alle Datensätze durchiteriert und geprüft, ob die gesuchte Person darin angegeben ist. Die entdeckten Standorte werden in einer Liste gespeichert und am Ende in dem Terminal ausgegeben.

Für den Export einer Anwesenheitsliste benötigt man neben dem Tag noch den Standort. Zuerst wird die Journaldatei eingelesen und anschließend wird durch alle Datensätze durchiteriert. Dabei wird

der Name zu jeder Person zum angegebenen Standort gesammelt und in eine CSV geschrieben die kompatibel mit MS Excel und LibreOffice ist.

Programmablauf

Die Erläuterungen zum Ablauf des Codes wird durch die folgende Grafik unterstützt.



Nachdem das Programm kompiliert und die Server des Programms gestartet wurden, wird auf die Interaktion eines Benutzers gewartet. Die Interaktion findet in der Regel über einen Browser statt. Theoretisch könnten andere Programme verwendet werden aber aus Gründen der Usability werden die meisten Benutzer die Anwendung über einen Browser nutzen.

Wenn nun ein Benutzer auf die Startseite des QR-Code Server geht, kann dieser unter n- verschiedenen Standorten wählen die durch die XML-Datei vorgegeben sind. Auf der Seite des Standorts wird ein QR-Code angezeigt den die Studierenden mit einem Smartphone einscannen können. In dem QR-Code ist eine URL enthalten, die einen zeitlich veränderlichen Token enthält sowie eine Standort ID.

Wenn nun ein Student zur Vorlesung erscheint, kann dieser mittels QR-Code App den Code einscannen und wird auf eine Anmeldeseite weitergeleitet. Als erstes wird geprüft, ob der Token des QR-Codes gültig ist. Wenn die Prüfung fehlschlägt wird ein Fehler geworfen und bei Erfolg wird eine weitere Prüfung angestoßen, die für die Ausfüllung der Eingabedaten wichtig ist. Wenn für den Benutzer bisher kein Eintrag für den Tag existiert, werden die Eingabefelder für Name und Adresse leer gelassen, sodass der Benutzer diese ausfüllen muss. Nach dem Versenden der Daten wird die Aktion als Anmeldung im Journal des aktuellen Tages eingetragen. Wenn für den Benutzer ein Eintrag existiert, werden die Angaben in die Eingabefelder automatisch eingetragen. Diese Aktion wird automatisch als Abmeldung gewertet und dementsprechend in der aktuellen Journaldatei eingetragen.

Anwenderdokumentation

Anmelden bei Ankunft:

Zum Anmelden rufen Sie die Startseite der Anwendung auf. (Default ist <https://localhost:8080/>) Dort befindet sich die Standortauswahl. Im Dropdown Menü wählen Sie den zu besuchenden Standort aus. Und bestätigen über "Senden".

Sie gelangen auf die zum ausgewählten Ort, entsprechende Seite. Auf dieser befindet sich ein QR-Code, welchen Sie einscannen können um zur Anmeldeseite des ausgewählten Ortes zu gelangen.

Hier erfolgt die Anmeldung, indem Sie Ihren Namen und Ihre Adresse eingeben und über den "Anmelden" Button bestätigen.

Abmelden beim Verlassen der Örtlichkeit:

Zum Abmelden rufen Sie wie auch bei der Anmeldung die Startseite auf. (Default ist <https://localhost:8080/>) Hier wählen Sie im Dropdown Menü den Standort aus, an welchem Sie sich befinden und vorher auch angemeldet haben und bestätigen über "Senden".

Sie gelangen auf die Seite mit dem QR-Code zur ausgewählten Örtlichkeit. Diesen scannen Sie und gelangen so auf die Abmeldeseite.

Falls Sie die Seite von Ihrer Anmeldung noch geöffnet haben, können Sie die Seite auch aktualisieren, sofern die Sitzung noch nicht abgelaufen ist.

Hier sehen Sie Ihre vorher eingetragenen Daten. Ihre Abmeldung bestätigen Sie dann über den Button "Abmelden" und Sie können die Örtlichkeit verlassen.

Verwendung des Analyse-Tools

Für die Verwendung des Analyse-Tools muss der Anwender ein Terminal öffnen und in das Verzeichnis navigieren, in dem sich das Analyse-Tool befindet. Dort angelangt kann dieser mit dem Befehl `go build` eine ausführbare Datei erzeugen, womit man das Programm anschließend starten kann oder man führt direkt `go run main.go` aus und startet das Programm ohne eine ausführbare Datei zu erzeugen.

Nachdem das Programm im Terminal ausgeführt wurde, können hier nun die spezifischen Kommandos des Tools ausgeführt werden. Die Liste der verfügbaren Kommandos und ihrer Parameter kann durch das Kommando `help` angezeigt werden. Für die Dokumentation wurde die Liste hier herein kopiert:

Kommando	Aktionsbeschreibung
help	Zeigt diese Hilfe an
select-day 21-12-2012	Wählt ein Datum für weitere Befehle aus
search-person max.mustermann	Sucht die Orte an dem sich eine Person am Tag aufgehalten hat
list-days	Zeigt die Tage an, an denen eine Anwesenheit protokolliert wurde
export-list Ort	Exportiert die Anwesenheitsliste für einen Ort in eine CSV-Datei
exit	Beenden

Für das Einlesen von Journaldateien ist es zuerst empfehlenswert, die Liste der verfügbaren Tage auszugeben, es sei denn, der Benutzer weiß, dass für den Tag ein Journal existiert. Der Tag wird durch das Kommando select-day MM-DD-YYYY ausgewählt. Nach dieser Auswahl können auch die Kommandos search-person und export-list benutzt werden. Für die genannten Kommandos werden zusätzliche Parameter wie Name oder Standort benötigt. Das Analyse-Tool lässt sich mit dem Kommando exit beenden.

Dokumentation des Betriebs

Voraussetzungen

Für die Inbetriebnahme der Anwesenheitsliste und des Analyse-Tools ist es erforderlich, dass Go auf dem Server oder dem Computer auf dem das Programm laufen soll, installiert ist. Die verwendete Version die in der Entwicklung verwendet wurde ist go1.17.5. Außerdem soll der Admin vor dem Start die Portnummern prüfen. Falls einer von beiden Ports für andere Programme verwendet wird, muss auf einen anderen Port ausgewichen werden.

Starten der Anwendung

Um die Anwendung auszuführen muss zuallererst der binäre Build mit dem Kommando "go build" über die Konsole gemacht werden.

Die eigentliche Anwendung kann danach mit dem Kommando "go run main.go" gestartet werden. Damit wird die Main-Methode der Anwendung ausgeführt. Beim Start können die Flags entsprechend der Vorgaben mit konfiguriert werden.

Testen der Anwendung

Das Testen der Anwendung erfolgt über die Eingabe des Kommandos "go test" über die Konsole.

Kommandos

Aktion	Kommando
Build binary	go build
Anwendung ausführen	go run main.go
Testen der Anwendung	go test

Flags

Flags	Description	Expected type
-portQR	port für die QR code Seite	int (Ganzzahl)
-portLogin	port für die An- und Abmelderseite	int (Ganzzahl)
-valid	Gültigkeitsdauer der Token in Sekunden	int (Ganzzahl)
-url	URL für das Erreichen der Webseite	string (text)

Mitgliederbeitrag

In diesem Abschnitt nennen die Gruppenmitglieder ihre einzelnen Aufgaben, die sie während der Implementierung bearbeitet haben.

Mit gewissen Überschneidungen wurde hauptsächlich an folgenden Teilen gearbeitet:

Beitrag von 3186523

- QR-Code Seite
- Https Verschlüsselung

Beitrag von 9008480

- Konfiguration (flags)
- An-Abmeldeseite
- Journal

Beitrag von 6196929

- Analyse-tool
- Token Generation
- XML Formating