Gruppe M: Digital Waitingroom

Dokumentation

Reutlingen, 04.06.2018

|  |  |  |
| --- | --- | --- |
| Nummer | Gruppenmitglieder (Namen) | Matrikelnummer |
| 1 | Marius Bauer | 752902 |
| 2 | Aberan Sivalingam | 752957 |
| 3 | Tobias Baader |  |

Inhaltsverzeichnis

[Einleitung 2](#_Toc518666338)

[Zielerklärung 2](#_Toc518666339)

[Lösungskonzepte / Design 2](#_Toc518666340)

[Ursprünglicher Zeitplan 3](#_Toc518666341)

[Rückblick 3](#_Toc518666342)

[Final Statusbericht (Marius) 3](#_Toc518666343)

[Inhaltliche Ergebnisse (Marius) 3](#_Toc518666344)

[Zusammenfassung 3](#_Toc518666345)

[Literatur 3](#_Toc518666346)

[Anhänge 3](#_Toc518666347)

# Einleitung

Obwohl der Patient einen Termin vereinbart hat, kommt es vor, dass er dennoch vor Ort nochmal warten muss. Unser Ziel ist es, sowohl den Patienten und den Ärzten Stress und Zeit abzunehmen, indem wir eine Anwendung schreiben, welches dem Patienten zeigt, wie lange er noch Zeit hat, bis er zu seinem Termin aufgerufen wird.

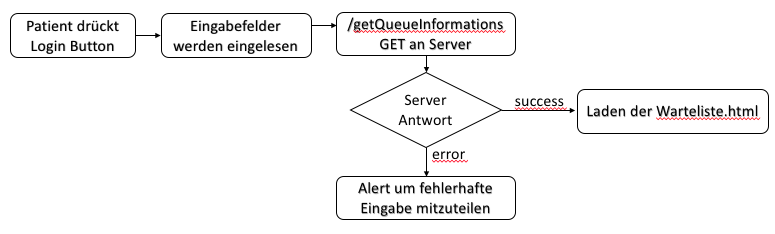
# Zielerklärung

Viele Patienten würden diese Wartezeit sinnvoller nutzen, als nur im Wartezimmer auf Ihren Termin zu warten. Durch unsere Anwendung, kann der Patient im Vorab sehen, ob es zu einer Verspätung bei seinem Termin kommt.

# Lösungskonzepte / Design

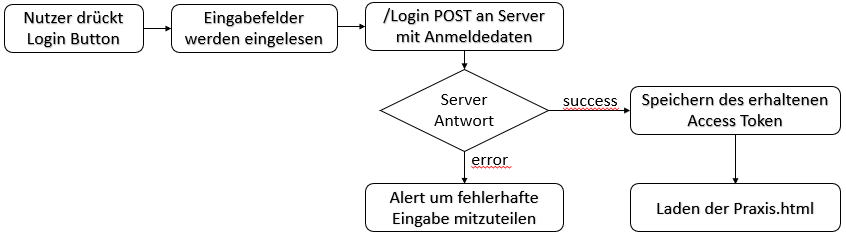
Für die Umsetzung haben wir geplant, Bootstrap für die Darstellung, MongoDB für die Datenbank und Node.js für den Server zu verwenden.

Patienten-Login:



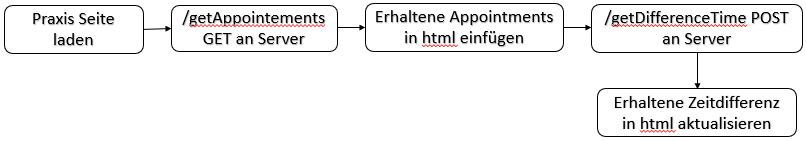
Sobald der Patient seinen Nachnamen und Geburtsdatum in den entsprechenden Feldern eingeben hat und auf den Login Button klickt, überprüft der Server, ob der Patient heute einen Termin hat. Falls dies der Fall ist, wird er auf die Warteliste.html weitergeleitet. Falls die Eingabe falsch oder kein Termin für heute vorgesehen ist, wird eine Fehlermeldung angezeigt.

Praxis-Login:



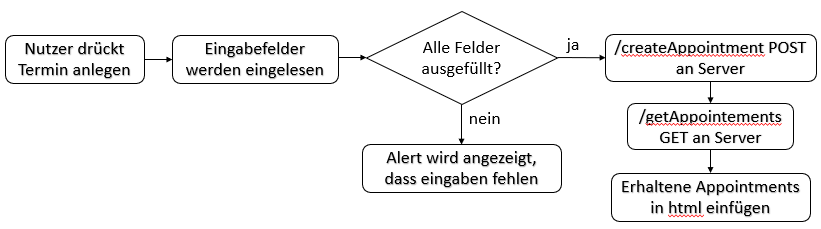
Der Praxis Login funktioniert ähnlich, wie der Patienten Login. Hier wird allerdings neben dem Usernamen ein Passwort verlangt und kein Geburtsdatum. Außerdem wird bei einer erfolgreichen Anmeldung ein von dem Server erhaltener Authorization Token als Cookie gespeichert und anschließend die Praxisseite geladen. Sind entweder das Passwort oder der Benutzername wird der Nutzer darauf hingewiesen.

Laden der Praxisseite:



Nach dem laden der praxis.html Seite wird eine Anfrage an den Server geschickt um alle Termine zu bekommen. Die erhaltenen Termine werden angezeigt, in dem sie dynamisch in die html eingefügt werden. Anschließend wird eine Anfrage nach der aktuellen Wartezeit gesendet. Auch die aktuelle Wartezeit wird dynamisch in die html eingefügt.

Termin Anlegen:



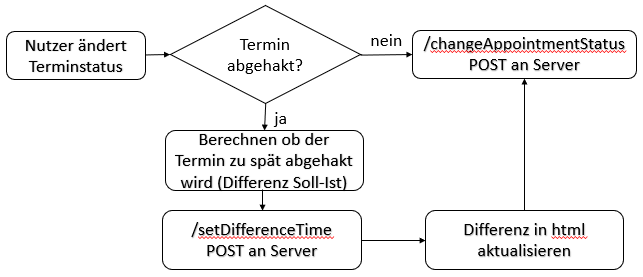
Drückt der Nutzer der „Termin Anlegen“ Button werden alle Eingabe Felder eingelesen. Es wird anschließend geprüft ob alle Felder ausgefüllt wurden. Ist dies nicht der Fall, wird der Nutzer darauf hingewiesen, dass alle Felder Pflichtfelder sind. Wurden alle Felder ausgefüllt, werden die ausgelesenen Informationen an als neuen Termin an den Server geschickt. Zuletzt werden die aktuellen Termine nochmal vom Server geholt und angezeigt.

Nutzer löschen:



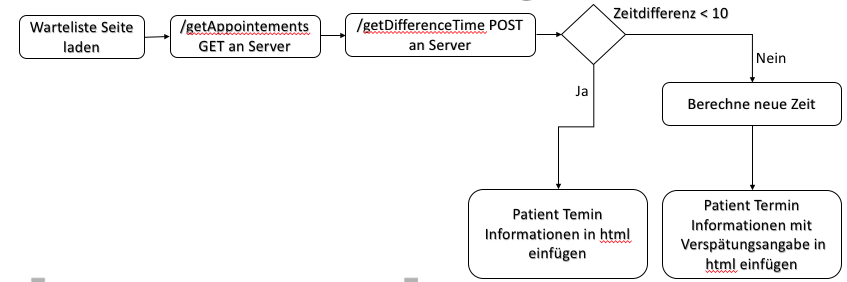
In der Termintabelle steht bei jedem Termin ein „löschen“ Button. Wird dieser gedrückt, wird der dazugehörige Termin aus der html Datei gelöscht. Anschließend wird eine Anfrage an den Server geschickt den Termin aus der Datenbank zu löschen.

Status Ändern:



In der Termintabelle steht bei jedem Termin eine Checkbox, mit der ein Termin als erledigt abgehakt werden kann. Ändert sich der Status der Checkbox, wird der zustand als der aktuelle Terminstatus an den Server geschickt. Wird der Status auf „true“ gesetzt, gilt der Termin als beendet und es wird die aktuelle Wartezeit berechnet. Die Wartezeit ergibt sich aus der Differenz der Sollzeit (Termin Anfang + Termin Dauer) und der Ist Zeit (Aktuelle Systemzeit). Nach der Berechnung wird die Wartezeit an den Server geschickt.

Patienten-Warteliste:



Nachdem der Patient erfolgreich seine Anfrage gesendet hat, wird er auf eine neue Seite weitergeleitet. Der Server nimmt dabei alle nötigen Informationen zu dem Patienten und überprüft, ob es zu einer Verspätung kommt. Falls dies der Fall ist, berechnet er, die neue Zeit und gibt diese zusätzlich auf der Seite aus.

# Ursprünglicher Zeitplan

Ursprünglich wurde die benötigte Zeit auf insgesamt 7 Wochen geschätzt. Dabei wurde berücksichtigt, dass alle Teammitglieder neben diesem Projekt gleichzeitig auch noch an anderen Projekten arbeiten werden. Somit war klar, dass man nicht mit acht Stunden am Tag rechnen kann. Bei der Zeitplanung wurde außerdem darauf geachtet, dass möglichst viele Aufgaben von parallel von verschiedenen Teammitgliedern bearbeitet werden können. So wurde zu Beginn des Projektes geplant, dass das Frontend und das Backend gleichzeitig und unabhängig voneinander entwickelt werden sollen. Erst ab der Schnittstelle zwischen Frontend und Backend wurden die Schritte sequenziell geplant.

# Rückblick

Risiken (Marius)

Während der Entwicklung:

* Zeitknappheit

Änderung von Plan oder Inhalt

Was lief gut / was schlecht?

|  |  |
| --- | --- |
| Lief gut | Lief schlecht |
| gemeinsame Zusammenarbeit | Durch Zeitknappheit, ist die Anwendung noch ausbaufähig |
| gegenseitige Unterstützung |  |
|  |  |

# Final Statusbericht (Marius)

# Inhaltliche Ergebnisse (Marius)

Usability Test:

UI Test:

Usability Test:

# Zusammenfassung

Die Entwicklung erfolgte im geplanten Zeitraum, auch wenn manche Aktivitäten länger und andere etwas kürzer als geplant brauchten.

Alle geplanten Funktionalitäten konnten umgesetzt werden. Es bedarf allerdings noch an einigen Stellen der letzte feinschliff. Insbesondere der Algorithmus zur Berechnung der Wartezeit muss überdacht werden und gegeben falls verbessert werden.

Die Anwendung ist nach unserer Meinung eine gute Idee, die in naher Zukunft sinnvoll genutzt werden kann.

# Literatur

Anwendungsentwicklung Tutorials, <https://www.w3schools.com>, letzter Zugriff, 06.07.18

# Anhänge

**Link zum GitHub:**

https://github.com/mx0c/E-Health