

Pflegeplaner



Pflege für Alle

1.1 Einleitung

Die Firma "Pflege für Alle", ein mobiler Pflegedienst, der hilfsbedürftigen Menschen Pflege und häusliche Unterstützung anbietet, hat uns gebeten, eine Software/App für sie zu entwickeln, um sie bei der Verwaltung zu unterstützen.

Die Software sollte für das Personal (Verwaltung und Pflege) zugänglich sein und den Pflegern ermöglichen, den Dienstplan einzusehen (der den Namen und den Ort jedes Patienten sowie das Datum und die Uhrzeit enthält, zu der die Pfleger bei ihm sein müssen), während das Verwaltungspersonal Dinge wie die Verwaltung des Dienstplans, das Hinzufügen oder Entfernen eines Mitarbeiters oder Kunden und die Überprüfung seiner Verfügbarkeit erledigen kann.

Die Gestaltung des Programms bleibt uns überlassen.

1.2 Rahmenbedingungen

- 8 Stunden Arbeitszeit
- Ausfälle von Kunden am selben Tag
- Skalierbarkeit, um mit dem Unternehmen zu wachsen
- Standortunabhängige Verfügbarkeit

2. Anforderungen

2.1 Funktionale Anforderungen

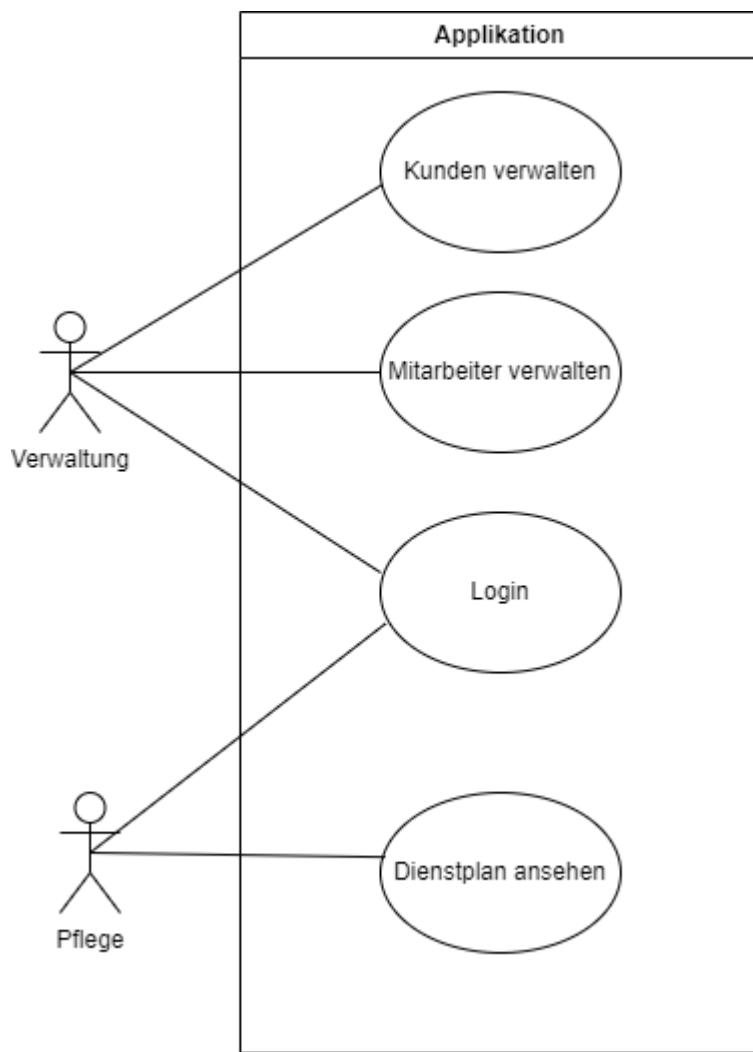
- (1) Alle Betreuer müssen ihre Tagespläne einsehen können: Die Pläne sollten Details wie den Namen des Klienten, den Ort, die Zeit und die Dauer des Besuchs enthalten.
- (2) Die Tagespläne müssen automatisch erstellt werden: Die Software sollte die Verfügbarkeit der Betreuer, den Standort der Klienten und die für jeden Besuch benötigte Zeit berücksichtigen, um die Tagespläne automatisch zu erstellen.
- (3) Es muss möglich sein, neue Kunden hinzuzufügen: Die Software muss es ermöglichen, neue Kundenprofile anzulegen, die Informationen enthalten wie: ob sie ambulant oder stationär sind, an welchen Tagen und wie oft am Tag sie unsere Pflegedienste benötigen.
- (4) Krankenbesuche müssen während des Tages berücksichtigt werden: Die Software muss in der Lage sein, die Zeitpläne in Echtzeit anzupassen, wenn sich ein Betreuer krank meldet, um sicherzustellen, dass alle Kunden weiterhin betreut werden.
- (5) Es muss möglich sein, neue Mitarbeiter hinzuzufügen: Die Software muss es ermöglichen, neue Mitarbeiterprofile anzulegen, die persönliche Angaben, Verfügbarkeit und Rolle enthalten.
- (6) Bei ambulanten Kunden muss die Reisezeit berücksichtigt werden: Die Software muss bei der Planung von Besuchen für ambulante Kunden die Reisezeit berücksichtigen, um sicherzustellen, dass die Pflegekräfte genügend Zeit haben, um von einem Kundenort zum anderen zu gelangen.
- (7) Für stationäre Klienten muss keine Reisezeit berücksichtigt werden: Da sich die stationären Klienten in derselben Einrichtung befinden, sollte keine Reisezeit in die Terminplanung einbezogen werden.
- (8) Warnung, wenn es zu wenige Betreuer für alle Klienten gibt: Die Software soll Warnungen ausgeben, wenn zu einem bestimmten Zeitpunkt nicht genügend Pflegekräfte für alle Klienten zur Verfügung stehen, damit gegebenenfalls zusätzliches Personal zugewiesen werden kann.
- (9) Es sollte eine Warnung ausgegeben werden, wenn es Betreuer ohne Klienten gibt: Die Software sollte eine Warnung ausgeben, wenn es Pflegekräfte gibt, die zu einem bestimmten Zeitpunkt keine Kunden zu betreuen haben, so dass gegebenenfalls zusätzliche Kunden zugewiesen werden können.
- (10) Die Kunden sollen mehrmals am Tag besucht werden können: Die Software soll mehrere Besuche pro Tag für Kunden ermöglichen, die mehr als einen Besuch durch eine Betreuungsperson benötigen. Das Planungssystem sollte flexibel genug sein, um mehrere Besuche durch verschiedene Betreuer zu ermöglichen, wobei für jeden Besuch genügend Zeit zur Verfügung steht, um den Bedürfnissen des Kunden gerecht zu werden.

2.2 Nichtfunktionale Anforderungen

- (1) Das Programm sollte zuverlässig sein, damit es nicht abstürzt oder Daten verliert: Die Software sollte mit einer robusten Architektur und einer gut getesteten Codebasis entwickelt werden, um sicherzustellen, dass sie zuverlässig arbeitet, ohne abzustürzen oder Daten zu verlieren. Außerdem sollte sie über angemessene Datensicherungs- und Wiederherstellungsmechanismen verfügen, um sicherzustellen, dass die Daten bei einem Systemausfall oder anderen unerwarteten Ereignissen nicht verloren gehen.

- (2) Das Programm sollte skalierbar sein, damit es im Laufe der Zeit mit einer wachsenden Zahl von Mitarbeitern und Kunden umgehen kann: Die Software sollte in der Lage sein, im Laufe der Zeit eine wachsende Zahl von Mitarbeitern und Kunden aufzunehmen, ohne die Leistung oder Stabilität zu beeinträchtigen.
- (3) Das Programm sollte eine Reaktionszeit von 0.2 Sekunden oder weniger haben, wenn es eine Aktion ausführt: Die Software sollte auf Geschwindigkeit optimiert sein, mit gut geschriebenem Code und effizienten Algorithmen, um sicherzustellen, dass sie Aktionen schnell ausführen kann.
- (4) Die Benutzerschnittstelle sollte einfach zu bedienen sein: Die Benutzeroberfläche der Software sollte benutzerfreundlich gestaltet sein, eine klare und intuitive Navigation aufweisen und nur minimalen Schulungsaufwand für neue Benutzer erfordern.

3. Use Cases



(1) Das Element "Verwaltung" soll die Mitarbeiter der Verwaltung im Unternehmen repräsentieren. Sie haben Zugang zu bestimmten Vorgängen und Seiten innerhalb der Website, wie "Login", "Kunden verwalten" und "Mitarbeiter verwalten".

(2) "Pflege" steht für die Betreuer im Unternehmen. Sie können sich auf der Website einloggen, haben aber nur Zugriff auf den Dienstplan.

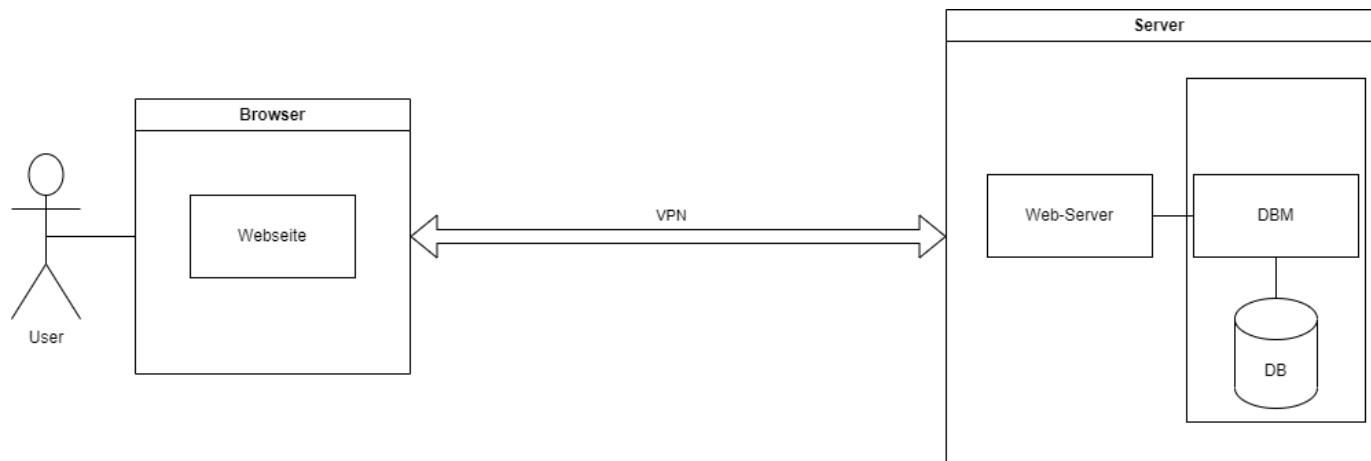
(3) "Login" ist der Bereich, in dem sich der Benutzer (entweder ein Mitarbeiter aus "Verwaltung" oder "Pflege") identifizieren muss, bevor er Zugang zur Website erhält.

(4) "Mitarbeiter verwalten" ist ein Bereich, zu dem nur die Mitarbeiter aus der "Verwaltung" Zugang haben. Hier kann der Benutzer einen anderen Mitarbeiter hinzufügen oder entfernen, sowie seine Krankmeldungen verwalten.

(5) "Kunden verwalten" ist dem Bereich "Mitarbeiter verwalten" sehr ähnlich, mit dem einzigen Unterschied, dass sich dieser Bereich auf die Patienten, und nicht auf die Mitarbeiter bezieht.

(6) "Dienstplan ansehen" bezieht sich auf den Dienstplan der Pfleger, auf den sie Zugriff haben. Hier können sie sehen, wer von ihnen wann und wo zu arbeiten hat.

4. Architektur



Die Software wird als Webapp entwickelt. Der Server wird mit Python und Flask entwickelt.

Server:

Genutzt wird WebFrameWork Flask mit dem Template System von JinJa und der SQL Schnittstelle Flask-SQLAlchemy

Die jeweiligen Seiten werden als Funktion aufgerufen und geben eine HTML Datei zurück

Gebraucht werden insgesamt 4 Templates: -Die Anmeldeseite womit man dann -> -zum Tagesplan kommt, wenn sich Mobile/Stationäre Pfleger anmelden -oder zur Verwaltung gelangt, wenn sich der Boss, oder Büroangestellte anmelden, wie z.B. die Sekretärin (in der Ansicht sieht man alle An/Abmeldungen, Krankschreibungen etc der Pfleger und Patienten -in dem 3. Template wird ein weiteres "kaskadiert" womit man bei dem Edit Button angelangt um die jeweiligen Einträge zu editieren

Man gelangt durch einen VPN zum Intranet der Firma, alle Mitarbeiter benötigen einen VPN Clienten den man kostenlos im App Store usw. installieren kann; danach ruft man die Seite der Firma auf im Browser. Falls im Laufe des Tages änderungen vorliegen, kümmert sich das Sekretariat darum.

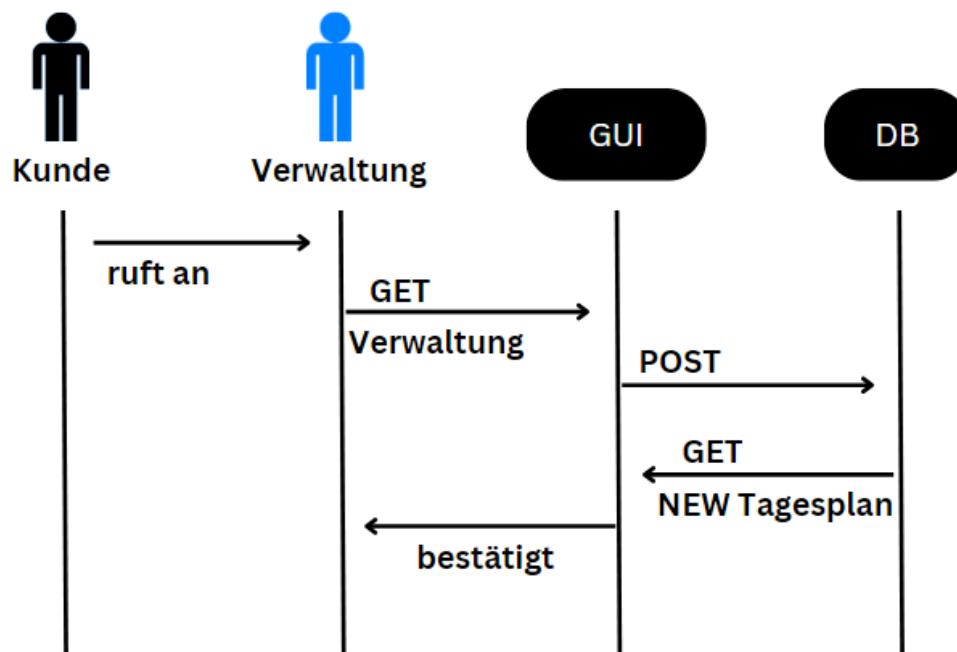
5. Geschäftsprozesse

1. Dienstbeginn Pflegepersonal

- Eigenen Namen im Menü auswählen (1F) und anfangen den Tagesplan abzuarbeiten (2F).
- Nach jedem Kunden schauen, wer der nächste Kunde ist.
- Wenn ein Kunde abgemeldet wird verschwindet er im Tagesplan.
- Wenn alle Kunden abgearbeitet sind, Feierabend.

2. Krankmeldung eines Kunden

Krankmeldung eines Kunden



- Kunde ruft an.
- Verwaltungsangestellte tragen Krankmeldung ein.
- Software berücksichtigt Krankmeldung.

3. Neuer Angestellter

- Verwaltungsangestellte/Chef tragen neuen Angestellten ein.
- Wahl der Rolle (Pfleger ambulant, Pfleger stationär, Verwaltung).

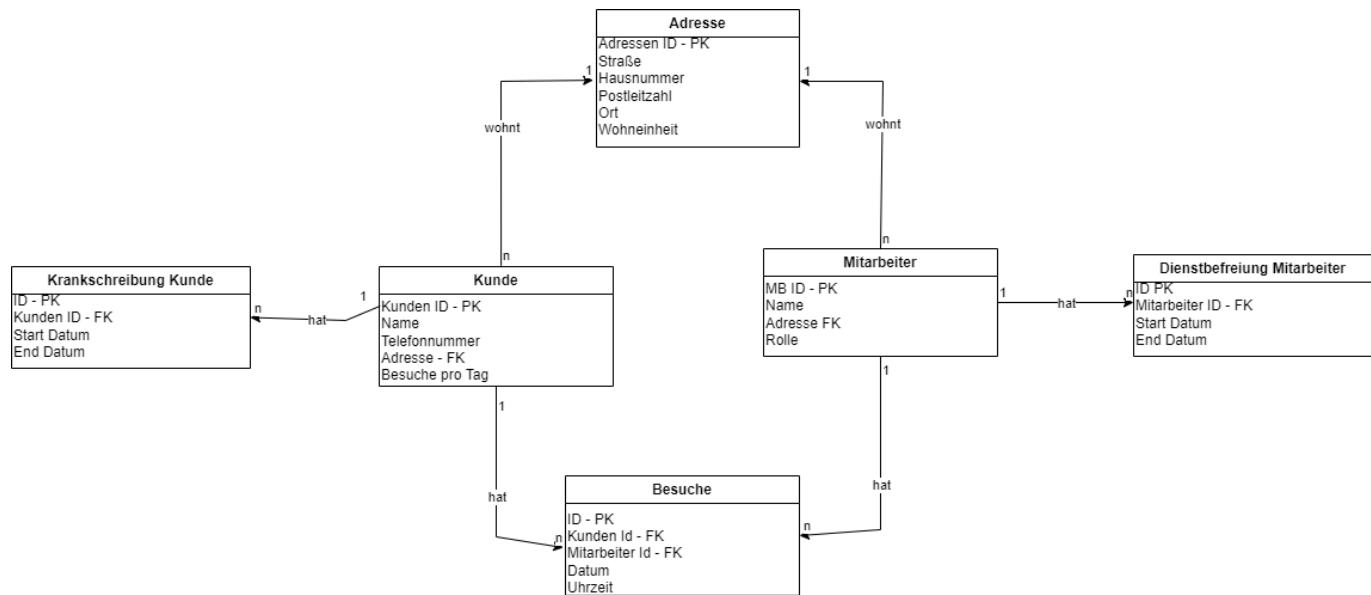
4. Neuer Kunde

- Verwaltungsangestellte/Chef tragen neuen Kunden ein.
- Wahl ob ambulant oder stationär.
- Wahl an welchen Tagen bedarf besteht.
- Wahl wie oft pro Tag bedarf besteht.

5. Dienstbeginn Verwaltung:

- Eigenen Namen im Menü auswählen (1F).
- Verwaltungsansicht (3F) wird angezeigt.

6. Datenmodell



Dies ist die Übersicht über den Aufbau der Datenbank.

Adresse

Enthält die Anschriften der Kunden und Mitarbeiter. Hat eine Fortlaufende ID als Private Key.

Mitarbeiter

Enthält die MitarbeiterID als Private Key, den Namen, Eine Adress ID als Foreign Key um die Adresse mit dem Mitarbeiter zu verknüpfen. Die Rolle legt die Funktion des Mitarbeiters fest.

Kunde

Analog zum Mitarbeiter mit den Unterschieden, dass es keine Rolle gibt sondern eine pflegeart die festlegt ob der Kunde stationär oder ambulant gepflegt wird.

Krankschreibung Mitarbeiter

Speichert einen Zeitraum in dem einem Mitarbeiter keine Kunde zugeteilt werden.

Krankschreibung Kunde

Speichert einen Zeitraum in dem ein Kunde keinem Pfleger zugeteilt wird.

Besuche

Speichert einen Zeitpunkt an dem ein bestimmter Pfleger bei einem bestimmten Kunden sein soll.

7. GUI-Design

Startseite

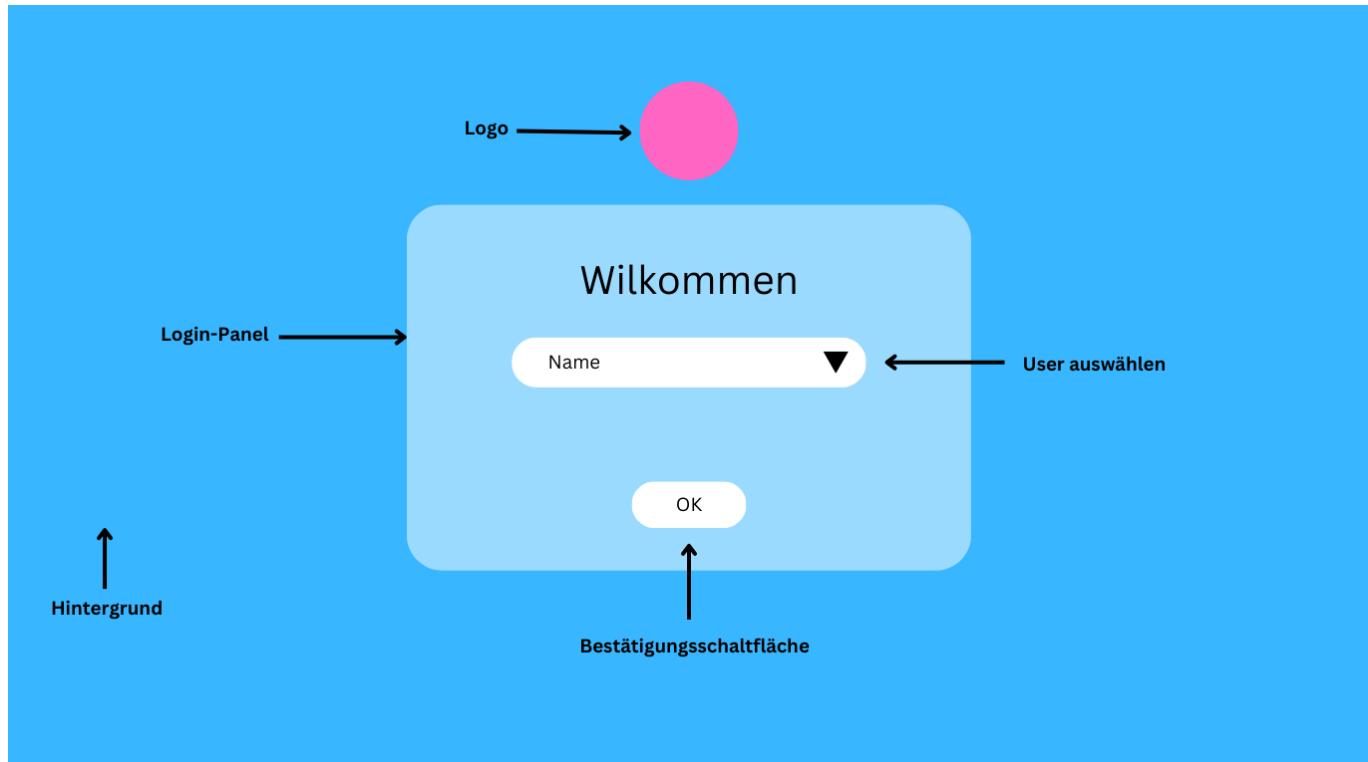


Abbildung Startseite

Dies ist die erste Seite, die der Benutzer sieht, wenn er die Website besucht. Die Seite besteht aus einem Hintergrundbild/-video, mit einem einfachen, minimalistisch aussehenden Anmeldepanel in der Mitte und dem Logo des Unternehmens darüber.

Der Benutzer wird mit einer "Willkommen"-Nachricht begrüßt und dann aufgefordert, seinen Namen in einem Textfeld einzugeben, wonach er auf die Schaltfläche "OK" klickt. Durch Klicken auf die Schaltfläche wird der Benutzer je nach Rolle, die seinem Namen im Unternehmen zugeordnet ist, zu [Tagesplan](#) oder [Verwaltungsansicht](#) umgeleitet.

Tagesplan

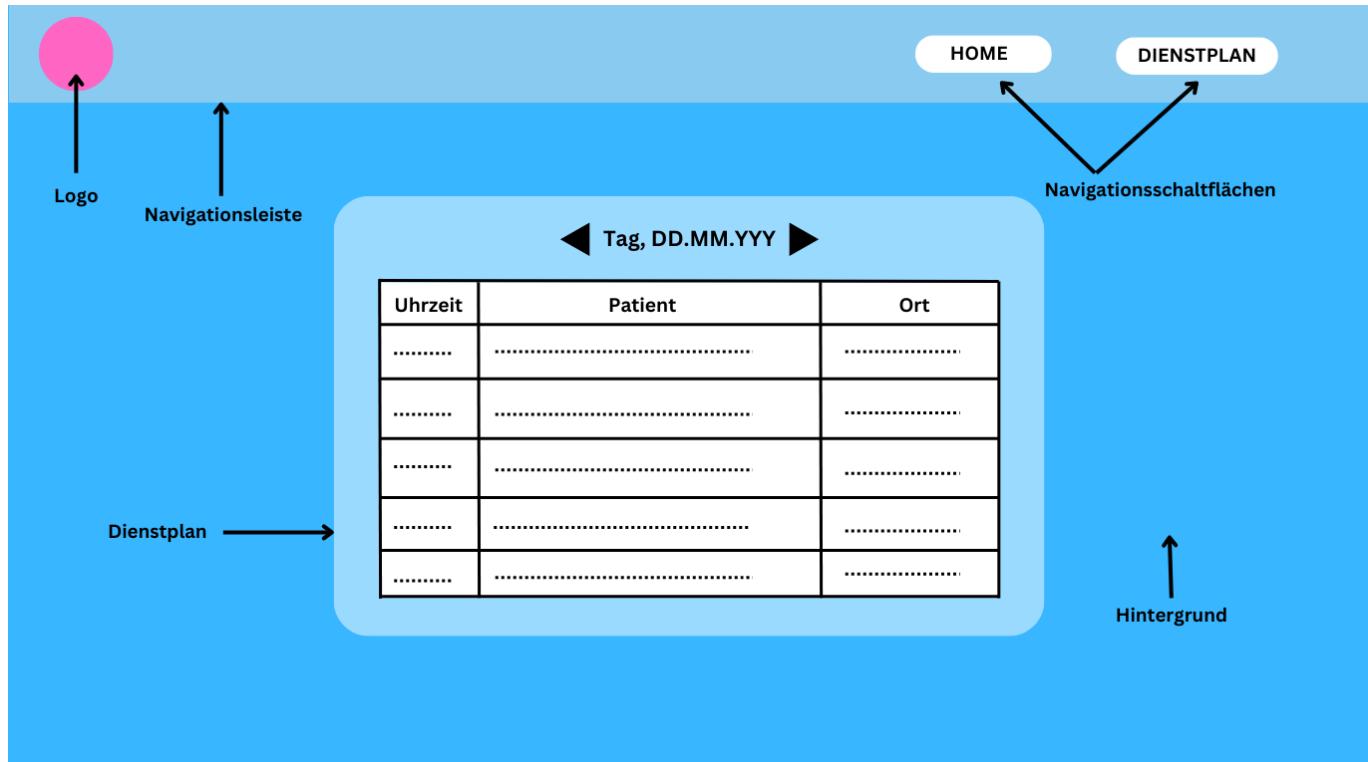


Abbildung Tagesplan

Diese Seite ist für sowohl Pfleger als auch Verwaltungsangestellte sichtbar. Hier sollte eine Navigationsleiste implementiert werden, die aus der Position des neuen Logos auf der linken Seite und zwei Navigationsknöpfen auf der rechten Seite besteht, genannt "Home" (welcher den Benutzer zur Login-Seite zurückleitet) und "Dienstplan" (wo wir uns gerade befinden). Die Seite besteht aus einer Tabelle, die den Dienstplan darstellt und Informationen wie den Namen des Patienten, seinen Standort sowie das Datum und die Uhrzeit enthält, zu denen der Pfleger den Patienten betreuen muss.

Verwaltungsansicht

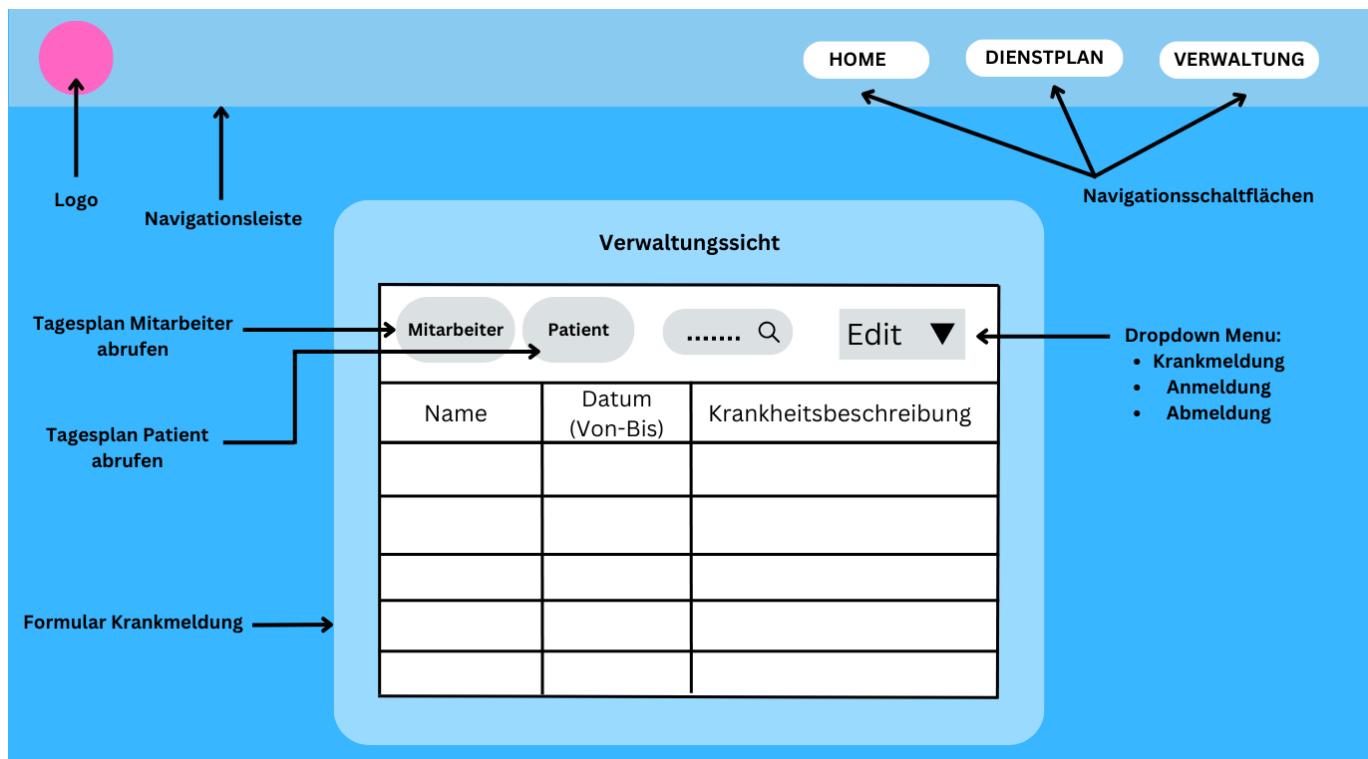


Abbildung Verwaltungsansicht

Zuletzt haben wir die Seite "Verwaltungssicht", die nur für Verwaltungsangestellte im Unternehmen sichtbar ist. Wie oben erwähnt, haben Verwaltungsangestellte Zugang sowohl zu den Seiten "Dienstplan" als auch "Verwaltungssicht".

Diese Seite hat das gleiche Layout wie "Dienstplan", jedoch mit einem zusätzlichen Button in der Navigationsleiste namens "Verwaltung", der den Benutzer zur Seite "Verwaltungssicht" weiterleitet. Sie besteht aus einem minimalistischen Panel, das es dem Benutzer ermöglicht, den Tagesplan für Mitarbeiter und Patienten zu sehen, einen Mitarbeiter oder Patienten hinzuzufügen oder zu entfernen und die Krankmeldungen zu verwalten, die per Telefon an die Verwaltungsangestellten gesendet werden. Es sollte auch eine Suchleiste geben, die es dem Benutzer ermöglicht, nach einem bestimmten Mitarbeiter oder Patienten zu suchen und deren Informationen zu überprüfen.

8. Implementierung

Nachverfolgung der funktionalen Anforderungen

1. Alle Betreuer müssen ihre Tagespläne einsehen können
2. Kankenbesuche müssen während des Tages berücksichtigt werden
3. Bei ambulanten Patienten muss die Reisezeit berücksichtigt werden
4. Die Kunden sollen mehrmals am Tag besucht werden können

Wenn die Betreuer eine Verbindung zum VPN haben oder sich im LAN befinden, können sie die Webseite aufrufen und sich mit ihren Daten einloggen.

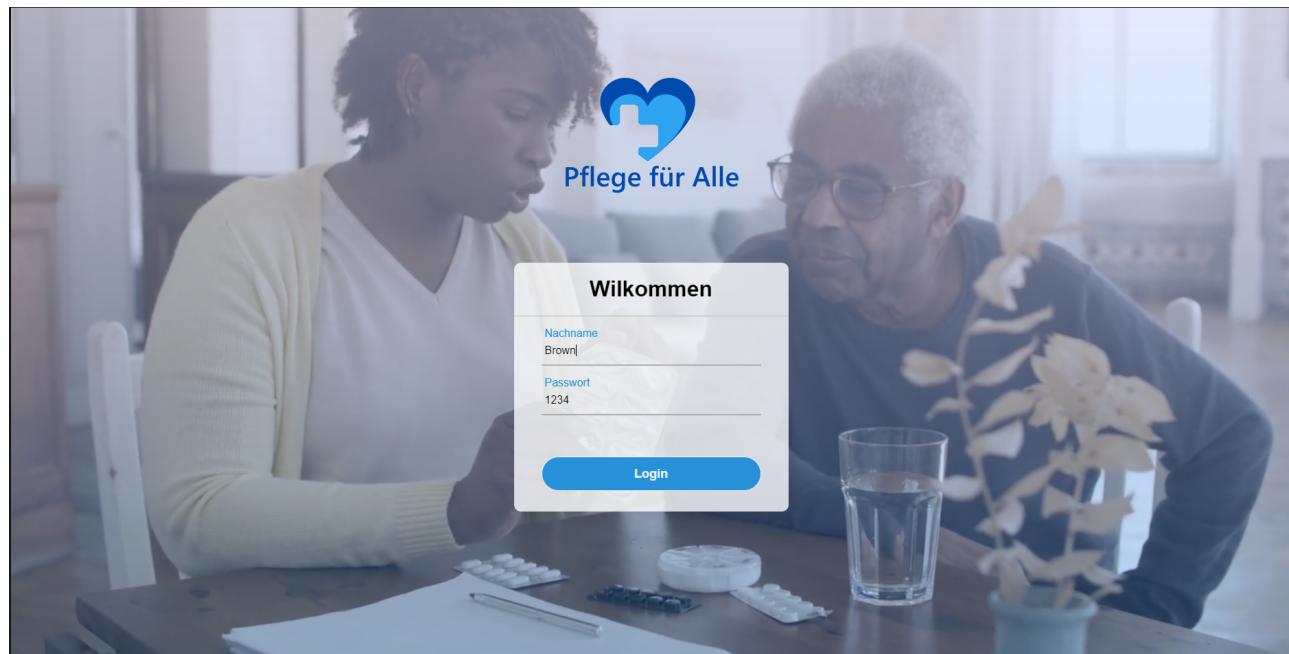


Abbildung Login

Danach wird der Tagesplan angezeigt.

The screenshot shows a software interface for 'Pflege für Alle'. At the top, there's a logo of a blue heart with a white cross inside, followed by the text 'Pflege für Alle'. To the right are buttons for 'Willkommen Brown', 'DIENSTPLAN' (which is highlighted in green), and 'LOGOUT'. Below this is a large, semi-transparent background image of a medical or office setting with a computer monitor and a keyboard.

In the center, a table titled 'Dienstplan aktualisieren' displays a daily schedule for a service provider named 'Brown'. The table has columns for 'Uhrzeit' (Time), 'Patient' (Patient), 'Tel.' (Phone), 'Rolle' (Role), 'Ort' (Location), and 'Wohnraum' (Residence). The data is as follows:

Uhrzeit	Patient	Tel.	Rolle	Ort	Wohnraum
08:30	Michael Johnson	555-3333	Ambulant	New Avenue 321, 98765 Newtown	3
09:30	David Wilson	555-5555	Ambulant	Example Boulevard 987, 54321 Exampleville	1
10:30	Sarah Brown	555-6666	Ambulant	Dummy Street 123, 87654 Dummy City	2
11:30	David Wilson	555-5555	Ambulant	Example Boulevard 987, 54321 Exampleville	1
12:30	Sarah Brown	555-6666	Ambulant	Dummy Street 123, 87654 Dummy City	2
13:30	David Wilson	555-5555	Ambulant	Example Boulevard 987, 54321 Exampleville	1
14:30	Oliver Anderson	555-8888	Ambulant	Sample Avenue 789, 65432 Samptown	3
15:30	Sophia Taylor	555-9999	Ambulant	Demo Lane 321, 87654 Demoville	2

Abbildung Tagesplan

Über den Button "Dienstplan aktualisieren werden Änderungen berücksichtigt". An den Uhrzeiten sieht man, dass die Reisezeit berücksichtigt wird. Auf dem Tagesplan sieht man, dass Kunden auch mehrmals am Tag besucht werden.

5. Die Tagespläne müssen automatisch erstellt werden

Das Modul "dienstplan_generator" erstellt die Tagespläne unter Berücksichtigung der Arbeitszeit, Abmeldungen von Kunden und Pflegepersonal, Bedarf an Besuchen pro Tag.

6. Es muss möglich sein, neue Kunden hinzuzufügen

7. Es muss möglich sein, neue Mitarbeiter hinzuzufügen

Wenn die Verwaltungsangestellten eine Verbindung zum VPN haben oder sich im LAN befinden, können sie die Webseite aufrufen und sich mit ihren Daten einloggen

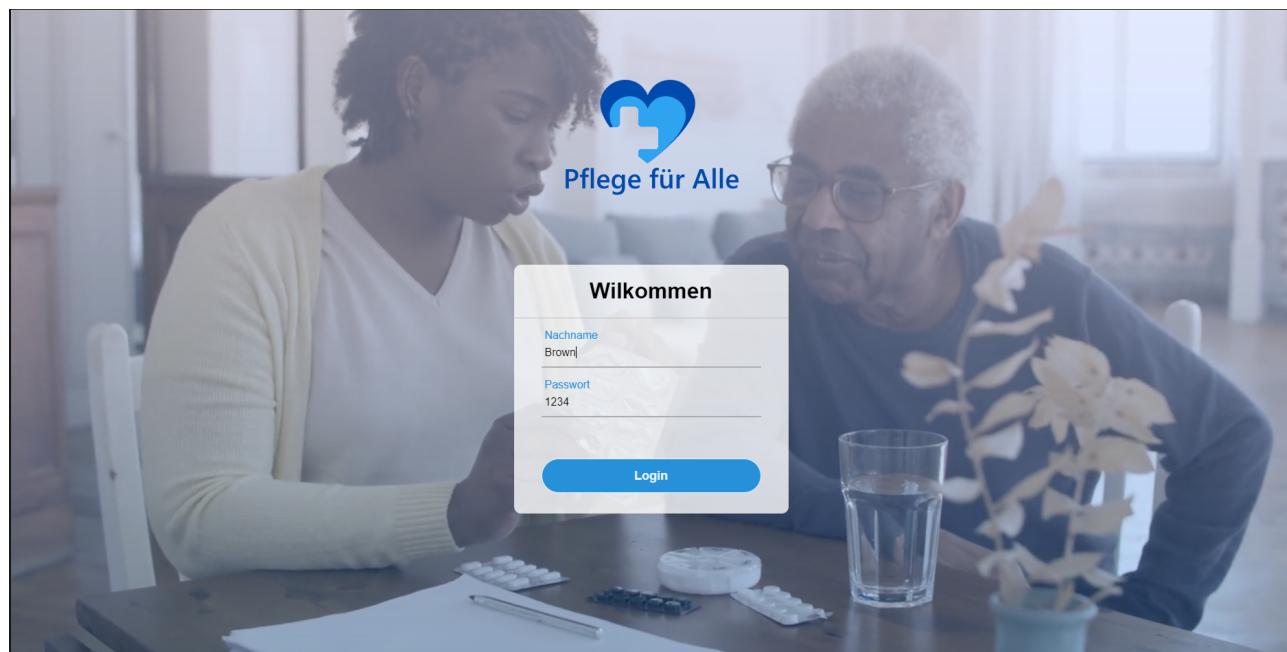


Abbildung Login

Über die Verwaltungsseite

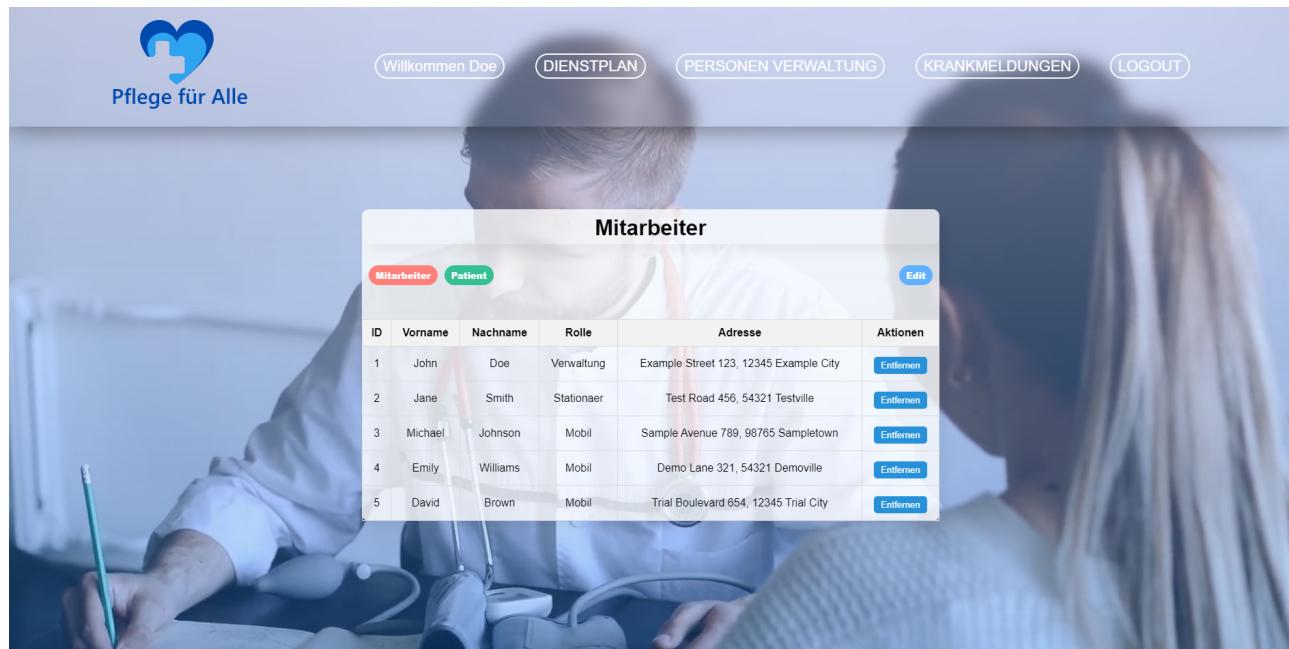


Abbildung Verwaltung

kommen die Verwaltungs angestellten zur Neuanmeldung von Patienten

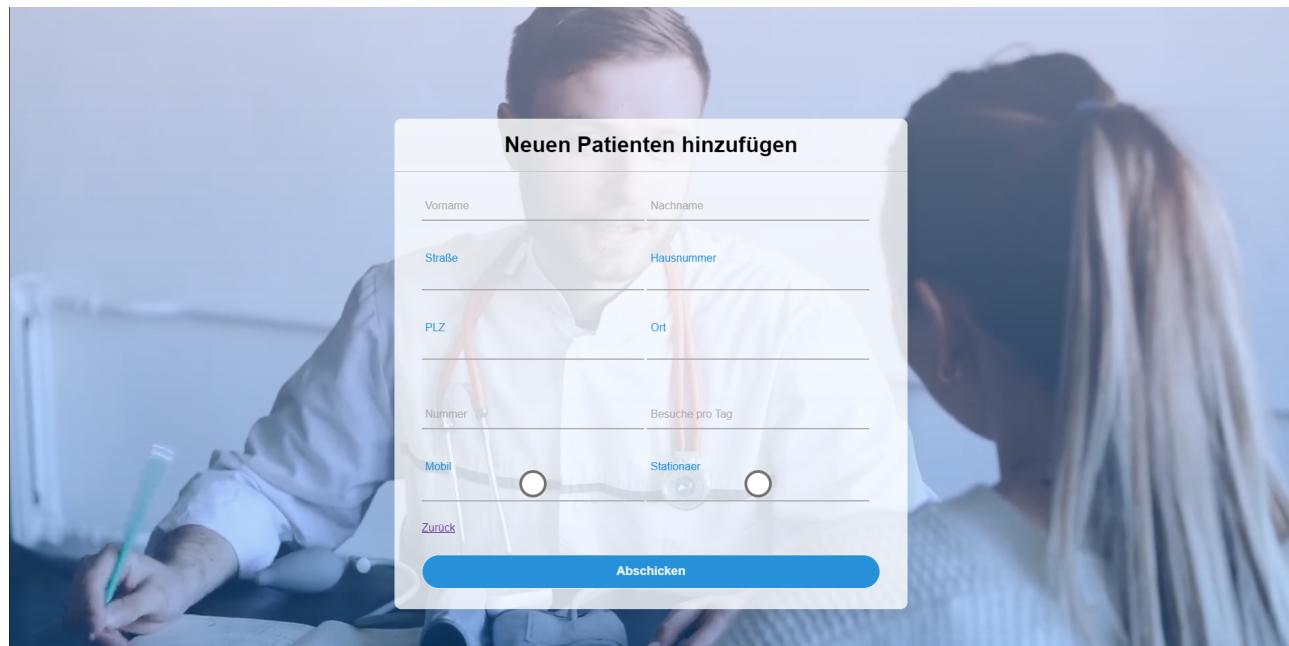


Abbildung Patienten hinzufügen

und Mitarbeitern

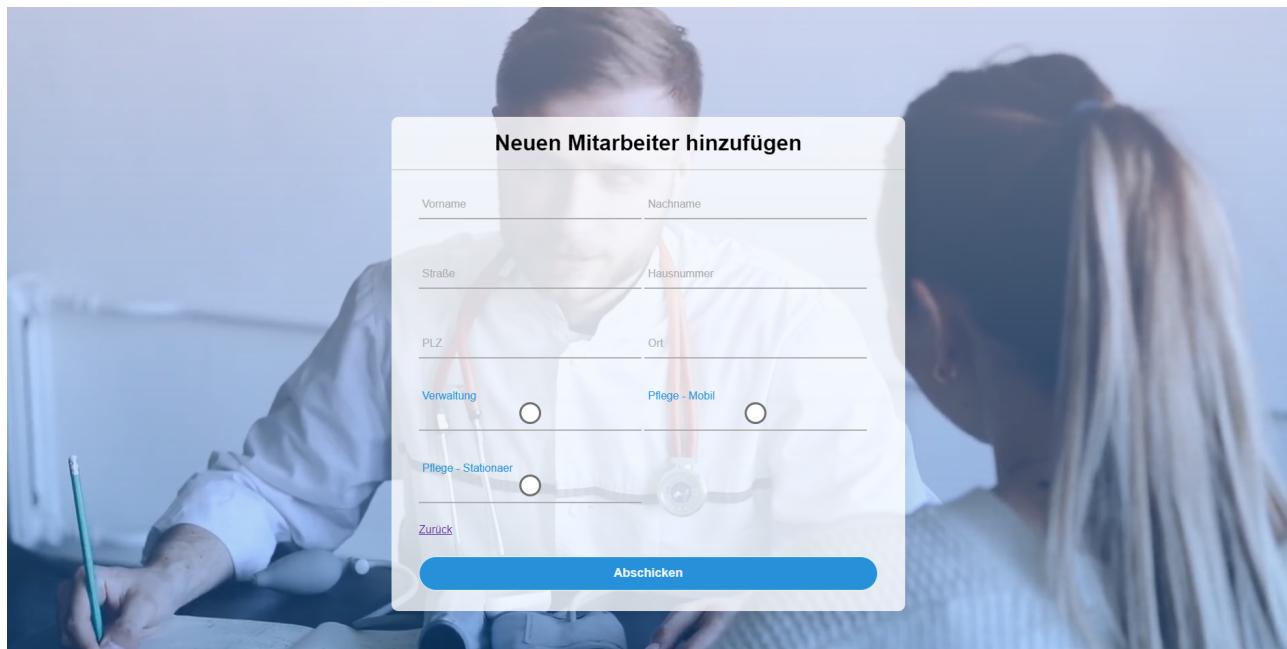


Abbildung Mitarbeiter hinzufügen

8. Für stationäre Klienten muss keine Reisezeit berücksichtigt werden

An den Uhrzeiten sieht man, dass bei stationären Patienten keine Reisezeit berücksichtigt wird.

Dienstplan aktualisieren					
Uhrzeit	Patient	Tel.	Rolle	Ort	Wohnraum
08:00	Bob Smith	555-5678	Stationär	Test Road 456, 54321 Testville	1
08:30	Jane Smith	555-2222	Stationär	Random Road 789, 65432 Randomville	1
09:00	Matthew Lee	555-7777	Stationär	Test Road 456, 34567 Testville	1
09:30	Noah Jackson	555-5678	Stationär	Random Road 789, 65432 Randomville	1
10:00	Bob Smith	555-5678	Stationär	Test Road 456, 54321 Testville	1
10:30	Jane Smith	555-2222	Stationär	Random Road 789, 65432 Randomville	1
11:00	Matthew Lee	555-7777	Stationär	Test Road 456, 34567 Testville	1
11:30	Noah Jackson	555-5678	Stationär	Random Road 789, 65432 Randomville	1
12:00	Bob Smith	555-5678	Stationär	Test Road 456, 54321 Testville	1
12:30	Jane Smith	555-2222	Stationär	Random Road 789, 65432 Randomville	1
13:00	Matthew Lee	555-7777	Stationär	Test Road 456, 34567 Testville	1
13:30	Noah Jackson	555-5678	Stationär	Random Road 789, 65432 Randomville	1
14:00	Jackson Johnson	555-2222	Stationär	Test Road 456, 34567 Testville	1
14:30	Liam Anderson	555-7777	Stationär	Random Road 789, 65432 Randomville	1
15:00	Liam White	555-5678	Stationär	Test Road 456, 34567 Testville	1
15:30	Henry Davis	555-2222	Stationär	Random Road 789, 65432 Randomville	1

Abbildung stationärer Tagesplan

Offene funktionale Anforderungen

8. Warnung, wenn es zu wenige Betreuer für alle Klienten gibt

9. Es sollte eine Warnung ausgegeben werden, wenn es Betreuer ohne Klienten gibt

Beide Warnungen sind nicht implementiert.

Nachverfolgung der nichtfunktionalen Anforderungen

2. Das Programm sollte skalierbar sein, damit es im Laufe der Zeit mit einer wachsenden Zahl von Mitarbeitern und Kunden umgehen kann

Das Programm ist in der Größe der Datenbank nur durch den Speicher des Servers limitiert und bei der Anzahl der Endgeräte die Anfragen an die Anwendung senden gibt es von der Software ebenfalls keine Limitierung.

3. Das Programm sollte eine Reaktionszeit von 0.2 Sekunden oder weniger haben, wenn es eine Aktion ausführt

Die Webseite hat eine Antwortzeit von weniger als 0.2 Sekunden.

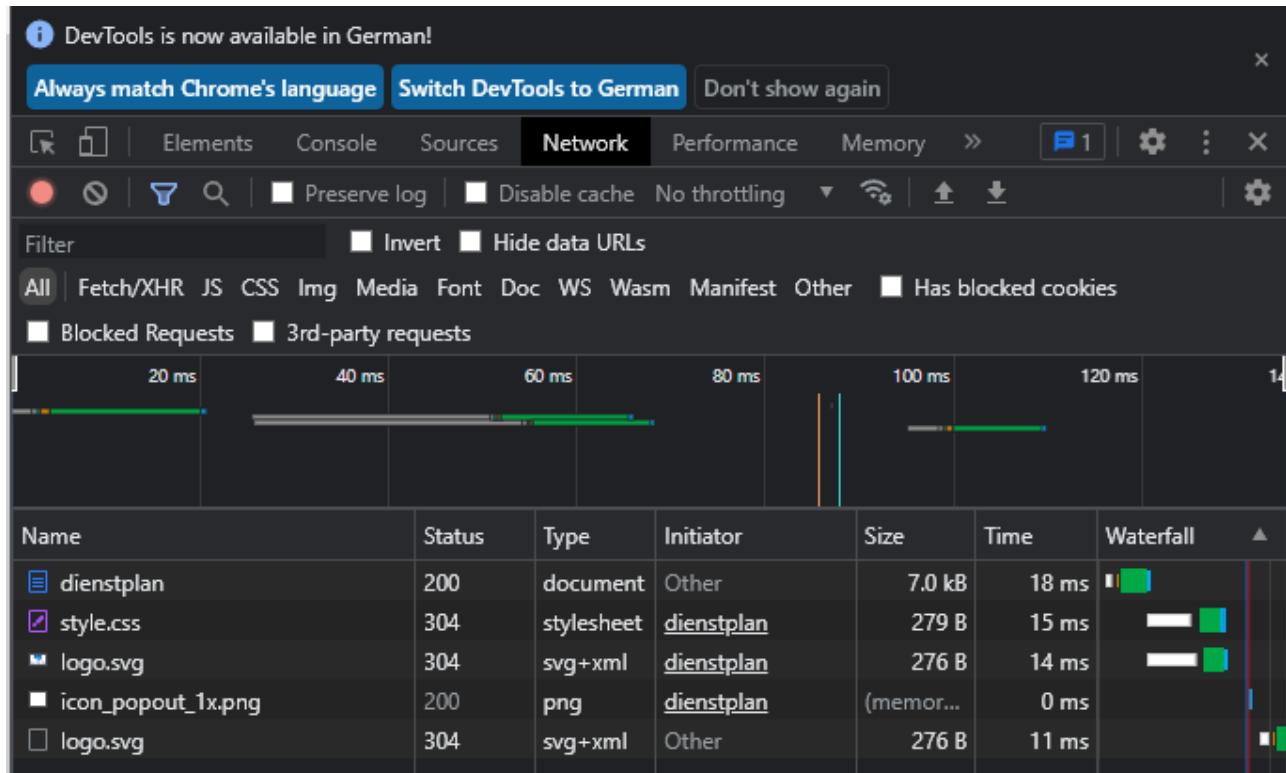


Abbildung Reaktionszeit

4. Die Benutzerschnittstelle sollte einfach zu bedienen sein

Wie in den Abbildungen oben zu sehen ist, sind die Oberflächen übersichtlich und alle Funktionen sind leicht zu finden.

Offene nichtfunktionale Anforderungen

1. Das Programm sollte zuverlässig sein, damit es nicht abstürzt oder Daten verliert

Es gibt keine Backups oder Synchronisation mit einer zweiten Instanz um den Server redundant zu betreiben.

Unterschiede zum Design

Es gab Anpassungen der Oberflächen um die Übersichtlichkeit zu verbessern.

Zum einen wurde das Suchfeld weggelassen. Zum anderen wurden die Spalten der Tabellen angepasst. "Krankmeldung" ist nicht mehr im Dropdownmenü "Edit", sondern in der Navbar. Der Login hat zusätzlich

noch ein Feld für ein Passwort bekommen. Es gibt beim Tagesplan einen Button zum aktualisieren.

9. Test

Unitests

Es gibt zwei Unitests.

Der erste Prüft ob die Startseite aufrufbar ist.

Der zweite prüft ob bei Seiten für die eine Authentifizierung vorausgesetzt wird, diese auch tatsächlich verlangt wird.

```
@pytest.mark.parametrize('path', (
    '/verwaltung',
    '/dienstplan',
))
def test_login_required(client, path):
    response = client.post(path)
    assert response.headers["Location"] == "/login"
```

Manuelle Testfälle

1. Login mit falschem Namen - muss fehlschlagen
2. Login mit Verwaltungsmitarbeiter - muss funktionieren
3. Login mit mobilem Pfleger - muss funktionieren
4. Login mit stationärem Pfleger - muss funktionieren
5. Zugriff auf Tagesplan mit Verwaltungsmitarbeiter - muss fehlschlagen
6. Zugriff auf Tagesplan mit mobilem Pfleger - muss funktionieren
7. Zugriff auf Tagesplan mit stationärem Pfleger - muss funktionieren
8. Zugriff auf Personenverwaltung mit Verwaltungsmitarbeiter - muss funktionieren
9. Zugriff auf Krankmeldungen mit Verwaltungsmitarbeiter - muss funktionieren
10. Neue Kunden anlegen mit Verwaltungsmitarbeiter - muss funktionieren
11. Neue Mitarbeiter anlegen mit Verwaltungsmitarbeiter - muss funktionieren
12. Neue Kundenkrankmeldung anlegen mit Verwaltungsmitarbeiter - muss funktionieren
13. Neue Mitarbeiterkrankmeldung anlegen mit Verwaltungsmitarbeiter - muss funktionieren

Quellen

Das Video, das beim Besuch der Website im Hintergrund abgespielt wird, enthält Clips, die von verschiedenen frei zugänglichen Plattformen bezogen wurden. Bei der Auswahl wurde darauf geachtet, dass alle Clips aus seriösen Quellen stammen, bei denen die Urheber der Inhalte ausdrücklich die Erlaubnis erteilt haben, ihre Arbeit frei zu verwenden, ohne dass eine Namensnennung oder irgendeine Form der Vergütung erforderlich ist. Links zu den Quellen:

- <https://www.pexels.com/video/a-woman-instructing-the-elderly-man-to-the-medicine-7517699/>
- <https://pixabay.com/videos/exercise-stretching-senior-elder-32937/>
- <https://pixabay.com/videos/blood-pressure-doctor-health-49815/>
- <https://www.pexels.com/de-de/video/frau-schreiben-arzt-gesundheit-4352136/>
- <https://pixabay.com/videos/hospital-medical-clinic-health-142535/>