

SWE - Praktikum 2

Sonntag, 23. April 2023 11:33

- Anfangs ganz allein zu den Patienten gefahren
- 40 Mitarbeiter beim 25. Firmenjubiläum
- Letztes Jahr -> noch ein Pflegeheim übernommen (+20 Mitarbeiter)
- Jetzt zu kompliziert

- Ein Mitarbeiter verbringt genau 30 Minuten bei jedem Kunden

Probleme:

- Mitarbeiter wurden schon zu falschen Patienten geschickt
- Einsatzplan ändert sich konstant - manche sind im Urlaub oder besuchen ihre Verwandten

Software:

- Software soll die Einsatzplanung übernehmen, sodass jeder ihrer Pflegekräfte jeweils seinen aktuellen Einsatzplan kennt. Also er weiß, bei welchem Kunden soll er wann sein soll.
- Programm sollte in der Lage sein, das Pflegepersonal automatisch auf der Grundlage seiner Verfügbarkeit und der Bedürfnisse der Kunden einzuteilen
- (?) Programm sollte die Fahrtzeit zwischen den Wohnungen der Patienten und dem Büro des Unternehmens berücksichtigen
- Programm sollte in der Lage sein, Änderungen im Zeitplan aufgrund von Urlauben, Krankenhausaufenthalten oder anderen Faktoren zu verarbeiten.
- Programm sollte einfach zu bedienen sein
- Programm sollte sicher sein und Kundendaten und Mitarbeiterinformationen schützen
- Programm sollte skalierbar sein, damit es im Laufe der Zeit mit einer wachsenden Zahl von Mitarbeitern und Kunden umgehen kann
- Programm sollte zuverlässig sein, so dass es nicht abstürzt oder Daten verliert (Internetverbindung? Oder Internal Server).
- Programm sollte Fehlermeldungen anerkennen und wiedergeben (Mehr Kunden als Angestellte -> Angestellten ohne Arbeit)

Ideen/Design:

1. Datenbankschema definieren :
 - Eine Tabelle "Kunden", um Informationen über jeden Kunden zu speichern, z. B. seinen Namen, seine Adresse und seine Kontaktinformationen.
 - Eine Tabelle "Mitarbeiter", um Informationen über jeden Mitarbeiter zu speichern, z. B. Name, Kontaktinformationen und Verfügbarkeitszeitplan.
 - Eine Tabelle "Besuche", um Informationen über jeden Besuch zu speichern, z. B. den besuchten Kunden, den dem Besuch zugewiesenen Mitarbeiter, das Datum und die Uhrzeit des Besuchs und den Status des Besuchs.
 - Weiter Tabellen
2. Datenbank erstellen:
 - Eine SQL-Datenbank installieren und konfigurieren, z. B. MySQL oder PostgreSQL.
 - Einen Datenbank-Client oder eine Befehlszeilenschnittstelle verwenden, um das Datenbankschema zu erstellen -> Wir könnten Herrn Davids fragen, ob wir Proxmox von WEB Praktikum verwenden und Apache installieren können.
3. Das Python-Programm erstellen:
 - Alle erforderlichen Python-Bibliotheken installieren, z. B. Flask für die Erstellung der Benutzeroberfläche und SQLAlchemy für die Datenbankinteraktionen.
 - Die Routen und Ansichten für die Flask-Anwendung definieren, z. B. eine Route zur Anzeige

einer Liste von Kunden und eine Ansicht zum Anlegen eines neuen Kunden.

- Die Modelle für das SQLAlchemy ORM definieren, z. B. ein Kundenmodell zur Darstellung eines Kunden in der Datenbank.
- Die Controller, um Benutzerinteraktionen zu behandeln definieren, z. B. das Anlegen eines neuen Kunden in der Datenbank oder das Aktualisieren der Informationen eines Kunden.
- Den Planungsalgorithmus, um die Mitarbeiter den Kundenbesuchen zuzuweisen definieren, wobei die Verfügbarkeit der einzelnen Mitarbeiter und die Reisezeit zwischen den Besuchen berücksichtigt werden.

4. Benutzeroberfläche erstellen:

- HTML, CSS und JavaScript verwenden, um eine einfache Benutzeroberfläche für das Programm zu erstellen, z. B. eine Seite zur Anzeige der Kundenliste und ein Formular zum Anlegen eines neuen Kunden.
- Flask-Vorlagen, um die Benutzeroberfläche in den Python-Code zu integrieren.

5. Programm testen:

- Unit-Tests verwenden, um einzelne Funktionen und Komponenten des Programms zu testen.
- Integrationstests benutzen, um das Programm als Ganzes zu testen, z. B. das Anlegen neuer Kunden und das Planen von Mitarbeiterbesuchen.

Anforderungsanalyse: Laut Kollege von mir: "Prof meinte, dass man ungefähr 3 Tage die Woche daran arbeiten muss" -> Wie in der art Pflichtenheft.

- Baumdiagramm -> Design: UI, Web-Aspekte, Programmieren, Datenbank...
- Sehr viel Text (Mindestens 8-10 Seiten -> Herr Davids mochte meine im WEB sehr, also da könnte ich ganz viel schreiben. Meine Programmier-Skills sind leider nicht die besten, aber hier bin ich "ok" :D)

Bezüglich HTML, CSS, JS und Python:

- Auf Stackoverflow habe ich das gesehen:
<https://stackoverflow.com/questions/29734208/how-can-i-connect-my-python-script-with-my-html-file>
- Hier geht es um wie man Python mit HTML verbindet, wann man am besten was benutzt etc (Ob die Daten dynamisch oder statisch sind, LAN oder WWW etc).

Sebastian Botezatu - OneNote, 23.04.2023