

Informationssystem für Tierklinik Projekt für Software Engineering INF202

Türkisch-Deutsche Universität

Projektbetreuer: DI. Ömer Karacan
Projektberater: Arş. Gör. Fulya Yenilmez

İdil Dilara Özmen e190503035@stud.tau.edu.tr
Dilara Serbest e190503037@stud.tau.edu.tr

Gliederung



- Einführung
 - Motivation
 - Ziele
- Lösungskonzept
- Stand der Technik
 - verwendete Programmiersprachen
 - Use Cases
 - Architekturüberblick
 - Controller Klassen
 - Systemtestanalyse
- Demo

Motivation

Als Tierhalterin freue ich mich, an diesem Projekt zur Entwicklung einer Tierklinik-App mitzuwirken. Diese App wird es uns leichter machen, Termine zu vereinbaren, auf die medizinischen Daten unserer Haustiere zuzugreifen und ihren Gesundheitszustand zu verwalten. Es ist eine bequeme Lösung, die die Betreuung unserer geliebten Haustiere verbessert und sicherstellt, dass sie die Aufmerksamkeit erhalten, die sie brauchen, wenn sie sie brauchen.



Abb1: Unsere Motivation für das Projekt

Die Ziele des Programms sind:

- Bereitstellung eines zentralen Systems zur Verwaltung von Patientenakten und Inventar für die Tierklinik.
- Erleichterung der Terminplanung und Zahlungsabwicklung für Patientenbesitzer.
- Verbesserung der Effizienz des Klinikbetriebs durch Reduzierung der manuellen Dateneingabe und Rationalisierung der Verwaltungsaufgaben.
- Gewährleistung der Sicherheit und des Datenschutzes von Patienten- und Zahlungsinformationen.

verwendete Programmiersprachen



Ich habe MySQL für die Datenbank, Java für das Backend und Java mit Hilfe von Android Studio in einer Anwendung mit geringem Zugang verwendet.

Ich hatte nicht viele Möglichkeiten für das Backend. Da Java eine der Sprachen ist, die ich am besten beherrsche, habe ich es in dieser Richtung eingesetzt.

Wir hatten nicht viel Ahnung von JavaFX, aber bei unseren Recherchen haben wir herausgefunden, dass JavaFX im Vergleich zu Webtechnologien wie HTML, CSS und JavaScript einen eher auf den Desktop ausgerichteten Ansatz zur Erstellung von Benutzeroberflächen bietet. Der Grund, warum ich Android Studio verwendet habe, war, eine kleine und einfache Implementierung zu erstellen, ohne das Projekt von seinem ursprünglichen Zweck abzulenken.

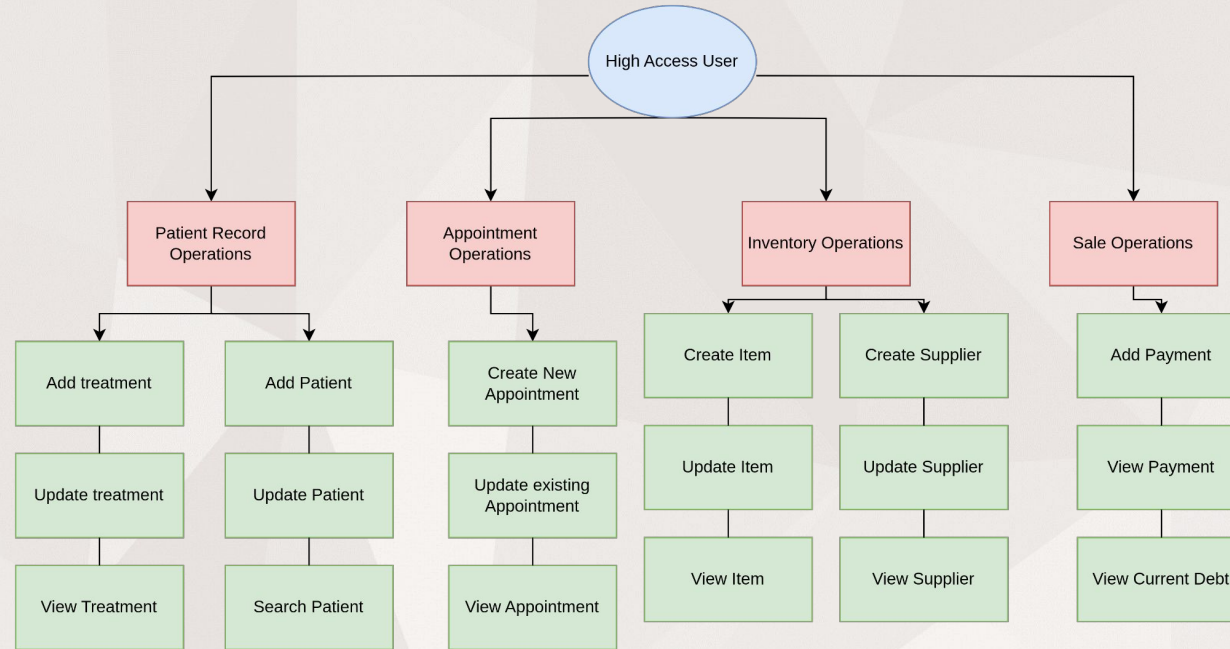


Abb2: High Access User Operations

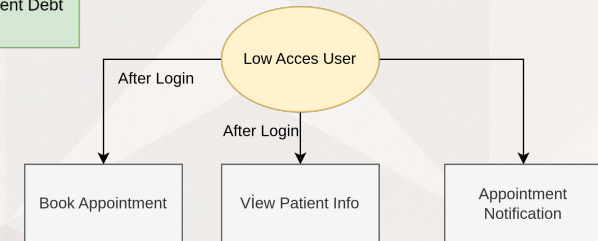


Abb3: Low Access User Operations

- 1) Der Benutzer mit hohem Zugang legt einen neuen Patientendatensatz im Programm an, indem er relevante Informationen wie Name, Alter, Tierart, Rasse, Krankengeschichte und Kontaktinformationen des Patienten eingibt, die später abgerufen und bei Bedarf aktualisiert werden können.
- 2) Patientenbesitzer Der Benutzer vereinbart über das Programm einen Termin für sein Haustier, wählt ein bevorzugtes Datum und eine Uhrzeit und erhält eine Terminbestätigung, die später eingesehen oder bei Bedarf storniert werden kann.
- 3) Der Benutzer mit hohem Zugang fügt einen neuen Artikel zum Inventar im Programm hinzu und gibt Details wie den Artikelnamen, die Beschreibung, die Menge und den Preis an, die später eingesehen und bei Bedarf aktualisiert werden können.
- 4) Der Patientenbesitzer kann die Krankengeschichte seines Tieres über das Programm einsehen und auf Details wie frühere Diagnosen, Behandlungen, Medikamente und Laborergebnisse zugreifen, die später bei Bedarf von einem Benutzer mit hohem Zugriff aktualisiert werden können.
- 5) Der Benutzer mit hohem Zugriffsrecht erstellt im Programm einen Verkaufsbericht, indem er einen Datumsbereich auswählt und eine Zusammenfassung der Verkaufsaktivitäten für den ausgewählten Zeitraum erstellt, einschließlich der verkauften Artikel, des erzielten Umsatzes und anderer relevanter Metriken.
- 6) Der Benutzer des Patienteninhabers bezahlt seine Rechnung über das Zahlungsportal des Programms, indem er die ausstehende Rechnung, die er bezahlen möchte, auswählt, seine Zahlungsinformationen sicher eingibt und eine Zahlungsbestätigung erhält, wobei das Programm den Rechnungsdatensatz des Patienten entsprechend der Zahlung aktualisiert.

Controller Klassen

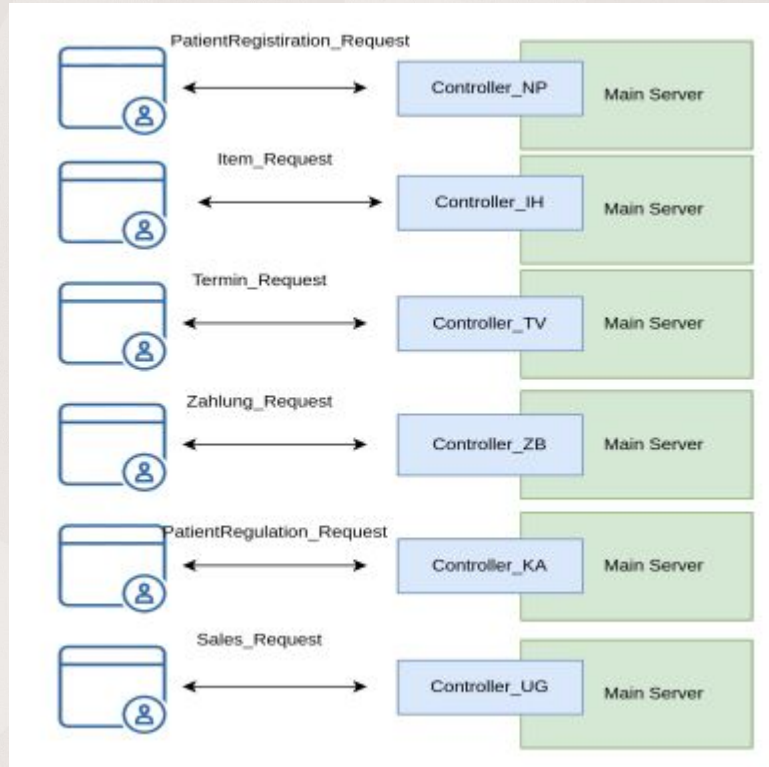


Abb4: Controller Klassen

PatientDataController: Verwaltet die Erstellung und Aktualisierung von Patientendatensätzen durch einen Benutzer mit hoher Zugriffsberechtigung, einschließlich relevanter Informationen wie Name, Alter, Tierart, Rasse, Krankengeschichte und Kontaktangaben.

AppointmentController: Verwaltet die Planung, Anzeige und Stornierung von Terminen für Haustierbesitzer und ermöglicht ihnen, bevorzugte Daten und Zeiten auszuwählen und eine Bestätigung zu erhalten.

InventoryController: Steuert das Hinzufügen und Aktualisieren von Bestandsartikeln, einschließlich Details wie Name, Beschreibung, Menge und Preis, durch einen Benutzer mit hohem Zugriff.

MedicalRecordController: Erleichtert die Anzeige und Aktualisierung der Krankengeschichte eines Haustiers, einschließlich früherer Diagnosen, Behandlungen, Medikamente und Laborergebnisse, die für den Tierbesitzer und einen Benutzer mit hohem Zugriffsrecht zugänglich sind.

SalesReportController: Erstellt eine Zusammenfassung der Verkaufsaktivitäten für einen ausgewählten Datumsbereich, einschließlich verkaufter Artikel, Einnahmen und anderer Metriken, für einen Benutzer mit hohem Zugriff.

PaymentController: Erledigt den Zahlungsvorgang für ausstehende Rechnungen und ermöglicht es dem Tierhalter, die Rechnung auszuwählen, die er bezahlen möchte, die Zahlungsinformationen sicher einzugeben und eine Zahlungsbestätigung zu erhalten.

Architekturüberblick

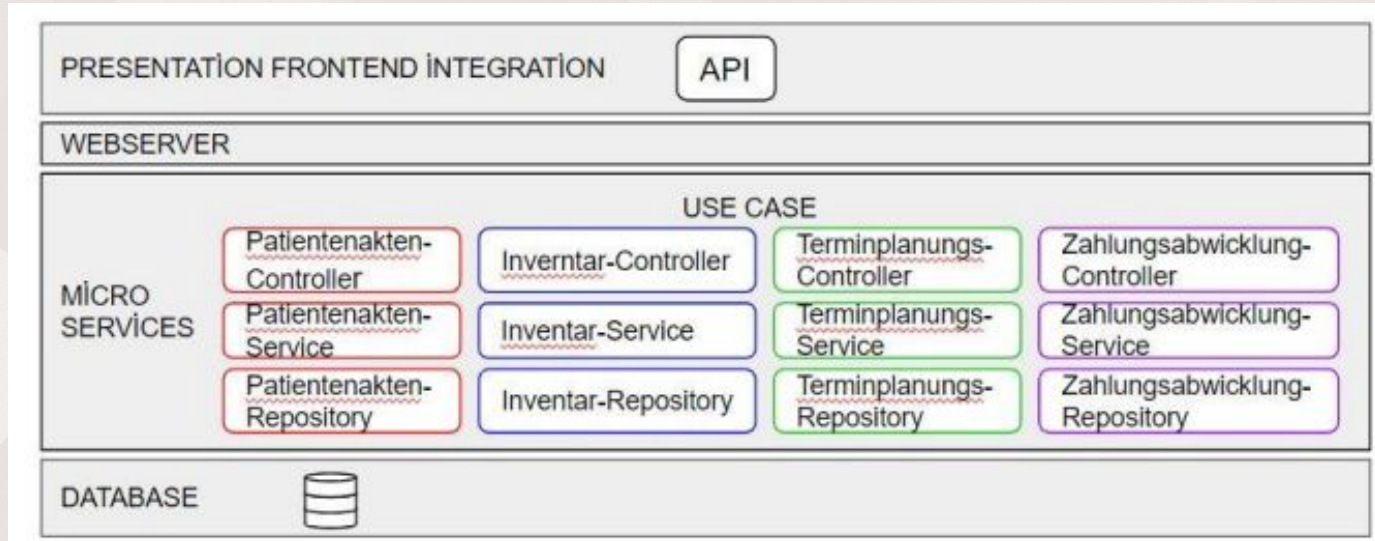


Abb5: Architekturüberblick für Informationssystem für Tierklinik

Demo



```
Select an Operation
1-Add new patient record
2-Update an existing patient record
3- View a patient
4-Create an inventory item
5-Update an existing item
6-View an item
7- Create a supplier

1
Chip ID:
asdf
Invalid input. Please enter a valid Chip ID.
Chip ID:
1234
Pet Name:
Çaki
Type:
Kedi
Breed:
Tekir
Colour:
3 renkli
Date Of Birth (yyyy-mm-dd):
2020-08-19
Age:
2
Owner's Name:
İdil Özmen
Contact Number:
05548339976
```

Ich kann keine vollständige Demo erstellen, weil unser Frontend nicht fertiggestellt , aber ich wollte zeigen, dass die Operationen in dem kleinen Code ausgeführt werden, den ich in der Main-Funktion definiert habe.

Abb6: Ein kleines Beispiel für den Backend-Betrieb

Low Access User Mobile App



Veterinary Information System

Chip ID

Password

LOG IN

Abb7: Login Activity

Welcome to Veterinary Info System

BOOK APPOINTMENT

VIEW PATIENT INFORMATION

LOGOUT

Abb8: Homepage Activity

Schedule Appointment

< January 1970 >						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SCHEDULE APPOINTMENT

Abb9: Appointment Activity

Vielen Dank für Ihre
Aufmerksamkeit!