Hochschule Fulda
University of Applied Sciences

Software Development Project - AI5094 (SoSe25)

Registered Care System – Individual Contribution

Simra Zaki Khan, fd0002065

Name:Simra Zaki Khan MtrNr.: 1542925

Requirement Specification

Hochschule Fulda
University of Applied Sciences

- **What I Did:** Wrote detailed plans explaining how the healthcare system should work
- **Why Important:** Created clear instructions for the team before building anything
- **Result:** Everyone understood what to build and how it should function

Requirement Specification

Hochschule Fulda
University of Applied Sciences

Business Requirements:

- Doctors can register new patients and manage their medical records.
- A unique Patient ID is automatically generated after the first diagnosis.
- Patients can log into the portal using their Patient ID to complete questionnaires.
- Doctors can track patient health progress through submitted questionnaire data.
- Doctors and shoemakers can assign and monitor patient diagnoses and questionnaire.
- Patients have limited access to their data, while doctors have full access to all patient records.

Design Document

Hochschule Fulda
University of Applied Sciences

To enhance the design documentation clarity, I incorporated wireframes that effectively explained the healthcare portal's user flows and process sequences. These visual representations helped document how patients navigate from diagnosis to questionnaire submission, and how doctors manage patient data, ensuring clear communication of the system's workflow logic to the entire development team.

For more details, click here: Design Document Link

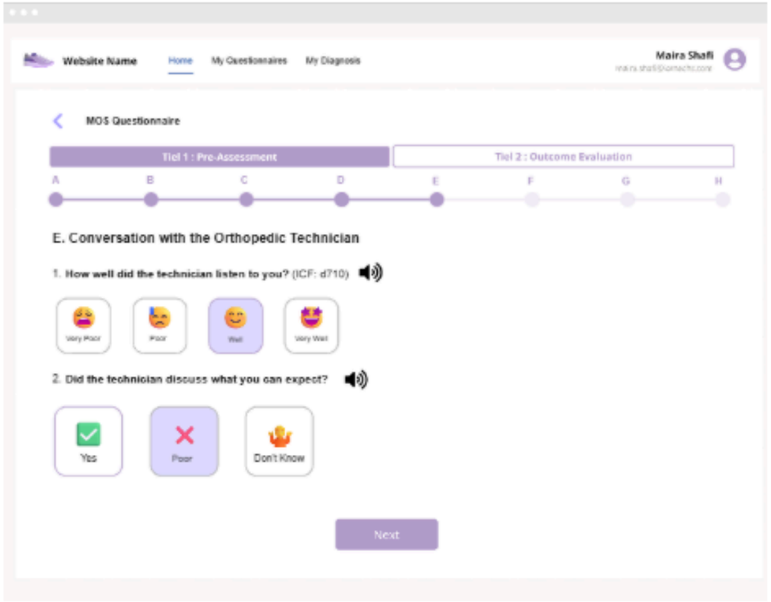
Design Document

Hochschule Fulda
University of Applied Sciences

2. Process Flow

Doctor's Workflow

1. Doctor creates a new Patient
2. Adds **Diagnosis 1** after first visit
3. System automatically assigns **Patient ID**
4. After **Diagnosis 1**, the system enables **Questionnaire 1**
5. Doctor adds **Diagnosis 2** after second visit
6. System enables **Questionnaire 2**



Patient's Workflow

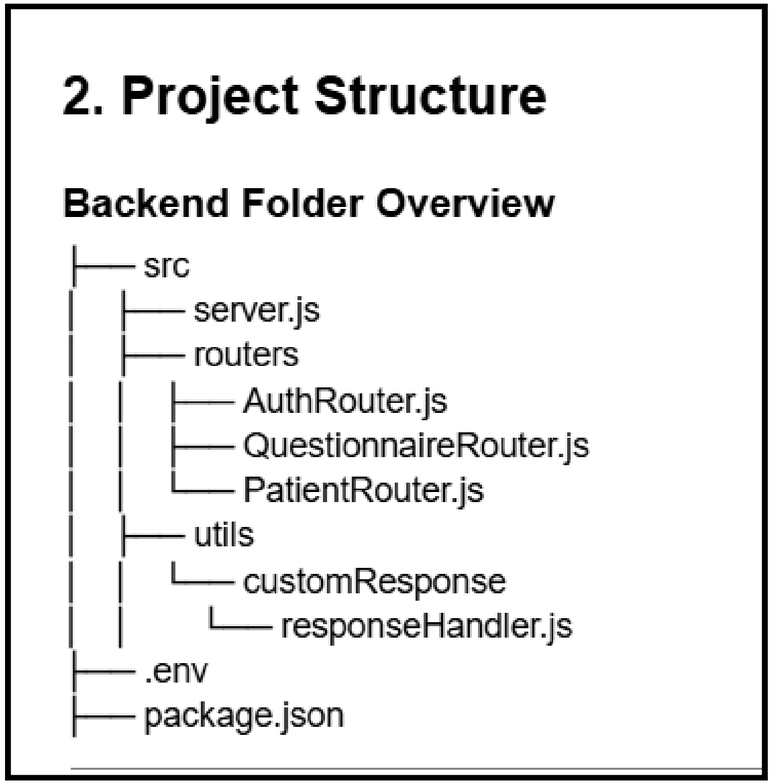
1. Patient receives **Patient ID** from doctor
2. Signs up using the Patient ID
3. Logs in and views:
    - Diagnosis records
    - Available questionnaires
4. Submits **Questionnaire 1** (after first visit)
5. Submits **Questionnaire 2** (after second visit)

CRUD API Development for Doctor & Patient Modules

Hochschule Fulda
University of Applied Sciences

Project Structure Understanding & Planning

- **What I Did:** First analyzed the complete project architecture to understand where each component belongs
- **Structure Mapping:** Identified where to place controllers, routers, and response handlers in the project hierarchy
- **Standards Implementation:** Followed established coding patterns and RESTful API conventions
- **Database Planning:** Discussed and decided which fields should be required vs optional for medical data



CRUD API Development for Doctor & Patient Modules

Getting Started with Backend Development

- **Learning Journey:** As I was new to backend development, I started with fundamental authentication features
- **First Implementation:** Built login and signup APIs to understand backend concepts and Node.js/Express.js
- **Foundation Building:** This helped me learn database connections, API structure, and response handling

CRUD API Development for Doctor & Patient Modules

Hochschule Fulda
University of Applied Sciences

Core Backend Development

- **What I Did:** Built the essential backend functions (Create, Read, Update, Delete) for medical data management
- **Why Important:** Enables doctors to add patients, patients to view their information, and everyone to update records safely
- T**echnical Implementation:**
  - Created organized folder structure (controllers, routers, utils)
  - Built separate routers for Authentication, Questionnaires, and Patients
  - Implemented custom response handlers for consistent API responses
- **Result:** Complete working backend system for healthcare management

Postman Collections for Backend Testing

Hochschule Fulda
University of Applied Sciences

- **What I Did:** Created automated tests to check if all APIs work correctly
- **Why Important:** Ensures the system works properly before patients and doctors use it
- **Result:** Executed test cases covering all medical scenarios and workflows

GitLab Collaboration & Branch Management

Hochschule Fulda
University of Applied Sciences

- **What I Did:** Organized code sharing and teamwork using GitLab platform
- **Why Important:** Multiple developers can work together without breaking each other's code
- **Result:** Smooth team collaboration and organized code development

QA Testing on Frontend & Backend Flows

Hochschule Fulda
University of Applied Sciences

Coming from a Quality Assurance background, testing was a natural strength and primary focus. Applied professional QA methodologies to ensure thorough system validation.

- **What I Did:** Tested all backend and frontend features to find and fix problems
- **Why Important:** Makes sure the healthcare system is reliable and error-free
- **Result:** High-quality strategies that handles medical data safely

Weekly Scrum Meetings Participation

Hochschule Fulda
University of Applied Sciences

- **What I Did:** Attended regular team meetings to discuss progress and problems
- **Why Important:** Keeps everyone updated and helps solve issues quickly
- **Result:** Better team communication and faster problem-solving

Links

Hochschule Fulda
University of Applied Sciences

 Requirement Specification Document
https://docs.google.com/document/d/1bEyUGGoqx65zAKcL--er-UlyLEDwxX03b6kg6skdeUs/edit?tab=t.0

 Design Document
https://docs.google.com/document/d/1jf5tdQ3yKmJUg_pAxXXT0IYXWcecrPc2kHGOpmZzDVg/edit?tab=t.0

 API Collection
https://gitlab.ietec-akademie.de/project-group-06/info/-/wikis/API_Collections