RECOVERY APP

The app for your injuries

Table of Contents

Analyses

If you have an injury, the first thing that you want is that it goes away. With my application called Recovery you will get a training exercise, a diet program and you can contact a physical therapist if there are any questions. It's easy to use, just register and give us information like your height and weight and you can start. When you have created an account, you can enter your injury and the app will generate a complete training schedule for you.

Requirements

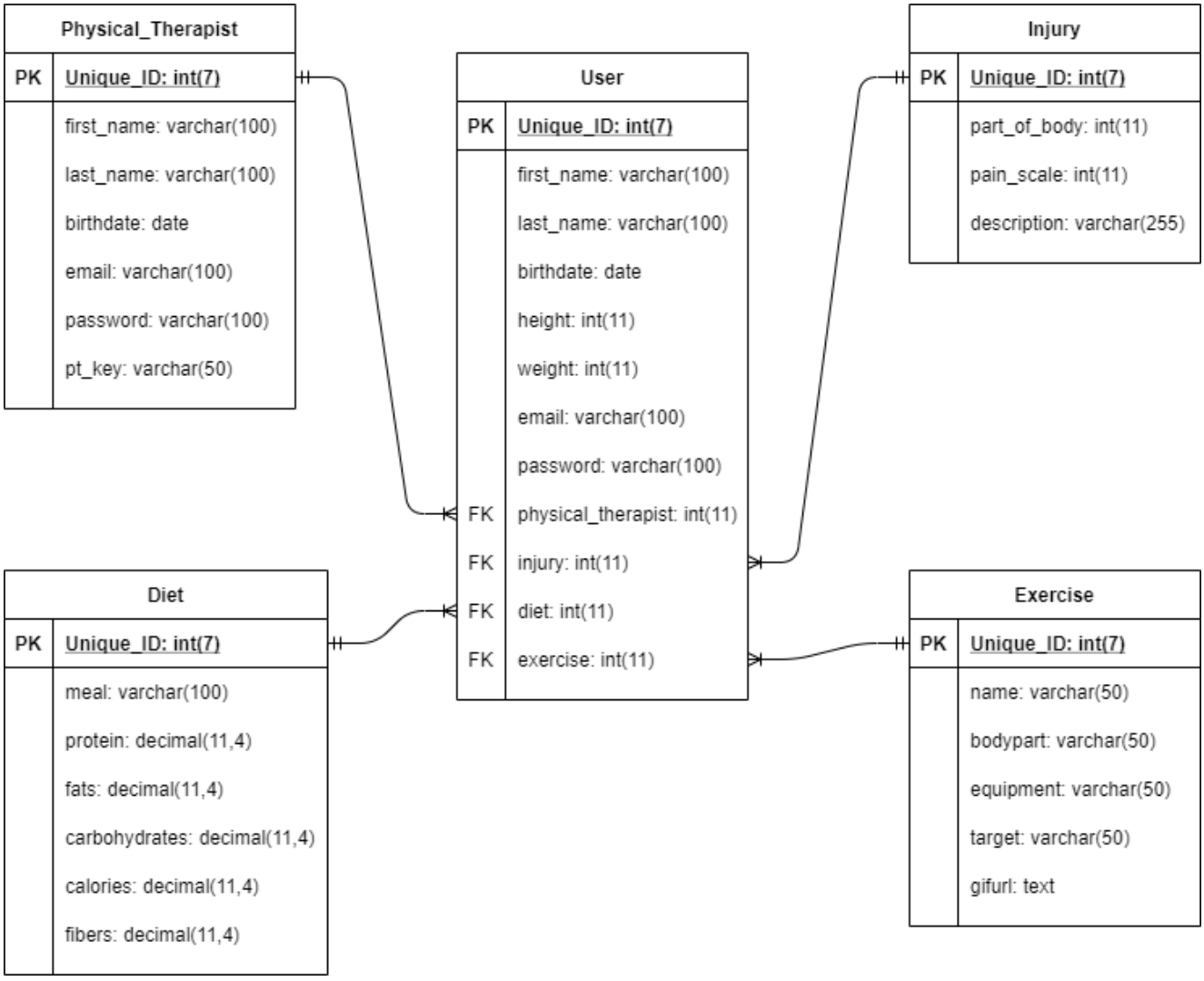
FUNCTIONAL REQUIREMENTS:

- FR01. A user must be able to create an account.
  - K1. If nothing is entered, the system gives an error.
  - B1. The password must be 8 characters long.
- FR02. A user must be able to login with his/her account.
  - K1. After trying logging in 3 times, the user must wait 2 minutes before being able to log in again.
- FR03. A user must be able to add a physical therapist to his account.
  - K1. A user can only add a physical therapist with a PT-number.
- FR04. A physical therapist must have a different view on the application than the user.
  - K1. The physical therapist can see a list of users that use his/her PT-number.
- FR05. A user must be able to select in which part of his body the injury is located.
  - K1. This must be shown in a list or in pictures.
- FR06. When injury is selected, a training schedule will appear on the user's account.
  - K1. This will be shown in a list, with the dates on top.
  - B1. This will take some time to load.
- FR07. A physical therapist must be able to give the user tips and answer questions.
  - K1. This is displayed in a chat box which contains the user and the physical therapist
- FR08. A user must receive a diet form if needed.
  - K1. This is shown in a list and shows all the needed nutrients.

NON-FUNCTIONAL REQUIREMENTS

- NFR01. The application will be shown in a unique corporate identity.
- NFR02. The application will be displayed in English or Dutch

Database model



Above you see a model of how the data will be stored in the database

Learning Outcome 1: Web Application

USER FRIENDLY

A good application needs to be user friendly, but what does user friendly mean? It means not only the responsiveness of the application but also the colors to accentuate the main topic of the application.

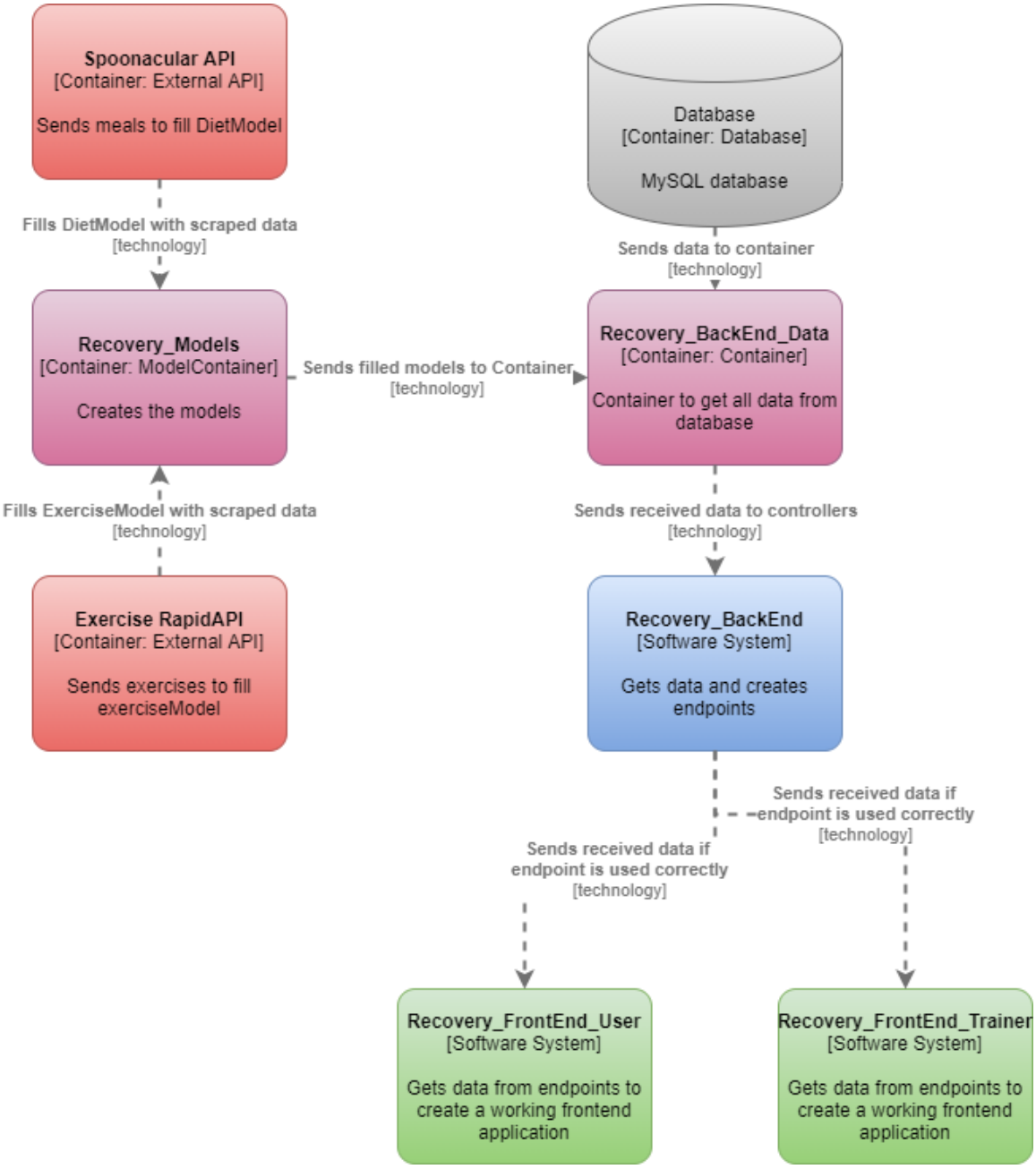
I have shown this by using one theme color in my whole application so it's not all messy with different colors. I have looked at different applications with the same physical therapist topic and most of the have used blue, so I have chosen blue as the main color for my application. I also have made the most of the application responsive for different devices.

FULL-STACK APPLICATION

I have built a full-stack application that includes the following:

- An ASP.NET Core web API backend with endpoints for every possible interaction in my application.
- 2 Frontends using React.js, which are divided by a user frontend and a physical therapist frontend.
- MySQL database
- 5 different components. These are: user component, physical therapist component, exercise component, diet component and injury component.
- 2 different external API's which are used to fill the diet component and the exercise component.

Dataflow


Spoonacular API
[Container: External API]
Sends meals to fill DietModel
Fills DietModel with scraped data
[technology]
Database
[Container: Database]
MySQL database
Sends data to container
[technology]
Recovery_Models
[Container: ModelContainer]
Creates the models
Sends filled models to Container
[technology]
Recovery_BackEnd_Data
[Container: Container]
Container to get all data from database
Fills ExerciseModel with scraped data
[technology]
Sends received data to controllers
[technology]
Exercise RapidAPI
[Container: External API]
Sends exercises to fill exerciseModel
Recovery_BackEnd
[Software System]
Gets data and creates endpoints
Sends received data if endpoint is used correctly
[technology]
Sends received data if endpoint is used correctly
[technology]
Recovery_FrontEnd_User
[Software System]
Gets data from endpoints to create a working frontend application
Recovery_FrontEnd_Trainer
[Software System]
Gets data from endpoints to create a working frontend application

Learning Outcome 2: Software Quality

TOOLING AND METHODOLOGY

Tests

I am doing Unit-Tests for every method in my data-layer to keep up with the consistency of my application. I also do an end-to-end test using cypress to test the compatibility of my frontends with my backends.

Security

As for the security part of my project I used the following:

- JSON Web Tokens (JWT) for authentication and authorization. A token is created when a user successfully logs in to the application and is saved in the session storage of the users' computer. When a user logs in successfully, the token is sent to the back end where the token will be validated.
- Password hashing using MD5, so the passwords of the users are secret for everyone that is authorized to user the database.

ORM

2 components are filled by the external APIs, so I use ORM in my database for my user component, physical therapist component and injury component. I am using it to manage the data that is send to my database and to prevent data redundancy.

Learning Outcome 3: CI/CD

DESIGN AND IMPLEMENTATION

I am using the following to fulfill this outcome:

- GitHub.
- GitHub actions.
- Docker.

GitHub

GitHub is a version management tool to help you continue working on your project if your PC just crashed and not losing any progress. I am using it to manage my project and documentation, so I don't lose anything.

GitHub actions

I am using GitHub actions for the CI part of this outcome. I chose GitHub actions because it's the most recommended to me by multiple students. The purpose of CI is controlling the pushed code, so it doesn't damage your current project. So, I am currently controlling my backend and my two frontends. I am also pushing my project to the Docker hub repositories.

Docker

Docker is fulfilling the CD part of this outcome. I mostly chose Docker because I didn't know any other deployment solutions and because it's recommended by students and teachers. I didn't know anything about Docker at first but with some help of some fellow students, I was able to create different containers and images for my project. These images are used to run the application instead of your own PC preventing a lot of configuration issues.

Learning Outcome 4: Professional

PROFESSIONAL MANNER

Feedback

I have tried to ask for feedback once a week, but it wasn't always possible to ask for feedback or I forgot to ask for a feed pulse point where I describe the given feedback. You can see the feed pulse points in the document called 'Feedpulse.pdf'.

Group project

Every day we had worked on the group project, we had a startup meeting to see where everyone was standing with their tasks. This was done to keep control on everything that had needed to be done for the requirements.

During the group project we had a lot of discussions about how we would be starting things or even about some choices. But we listened to everyone and their points on why we should choose their idea. That way we made choices where we struggled to get the answer.

After getting feedback from our stakeholders, we discussed the feedback and applied them to our plan to make priorities.

Learning Outcome 5: Agile

AGILE

Planning

At the beginning of the group project we made a Trello board to plan our project. Trello is a plan system site to create an agile way of planning your project and assign people to tasks. We split the requirements in the amount of sprints we had planned and worked our way trough the tasks. I also made a Trello board for my own project to get a view of what task I have to do.

Story points

While making the planning you need to set the priorities of what tasks your doing first. That is where story points come in. Story points are used to estimate the difficulty and amount of work is bound to a task. Most of the time the numbers in the Fibonacci sequence are used for the amount of story points.