

Recovery App

The app for your injuries



[Date]

[Company name]

[Company address]

# 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 schedule; 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 has to 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.

# Use-Cases

|  |  |
| --- | --- |
| Name | UC01: Add a User |
| Recap | When a user fills the registration form with all the information, an account will be created |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. User puts information in registration form. 2. System checks if all information is given. 3. System sends information to database. |
| Exceptions | 1. Not all information is given. Give error and go back to step 1. |
| Result | A new user is added |

|  |  |
| --- | --- |
| Name | UC02: Login as a User |
| Recap | When a user puts the correct details in the login form, he/she will be logged into the application |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. User puts information into Login form. 2. System checks if everything is filled. 3. Information is compared with the information in the database. 4. User is logged in |
| Exceptions | 1. Information user has put does not match with the information in the database. Give error and go back to step 1. |
| Result | User is logged in |

|  |  |
| --- | --- |
| Name | UC03: Adding a physical therapist |
| Recap | When a user gets his training schedule, he will be able to add a physical therapist. |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. UC06 2. User fills PT-number into form 3. PT-number is checked in database to find a physical therapist. 4. Physical therapist is added to user |
| Exceptions | 1. PT-number does not match with any physical therapists in database. Give error and go back to step 2. |
| Result | A physical therapist is added to a user. |

|  |  |
| --- | --- |
| Name | UC04: Physical therapists view |
| Recap | A physical therapist must see a list with all the users linked to the therapist. |
| Actors | Physical therapist, system, database |
| Assumptions |  |
| Scenario | 1. Physical therapist logs in. 2. System receives all the users linked to the PT. 3. System displays all the users in a list form |
| Exceptions |  |
| Result | PT’s view is a list with all the linked users |

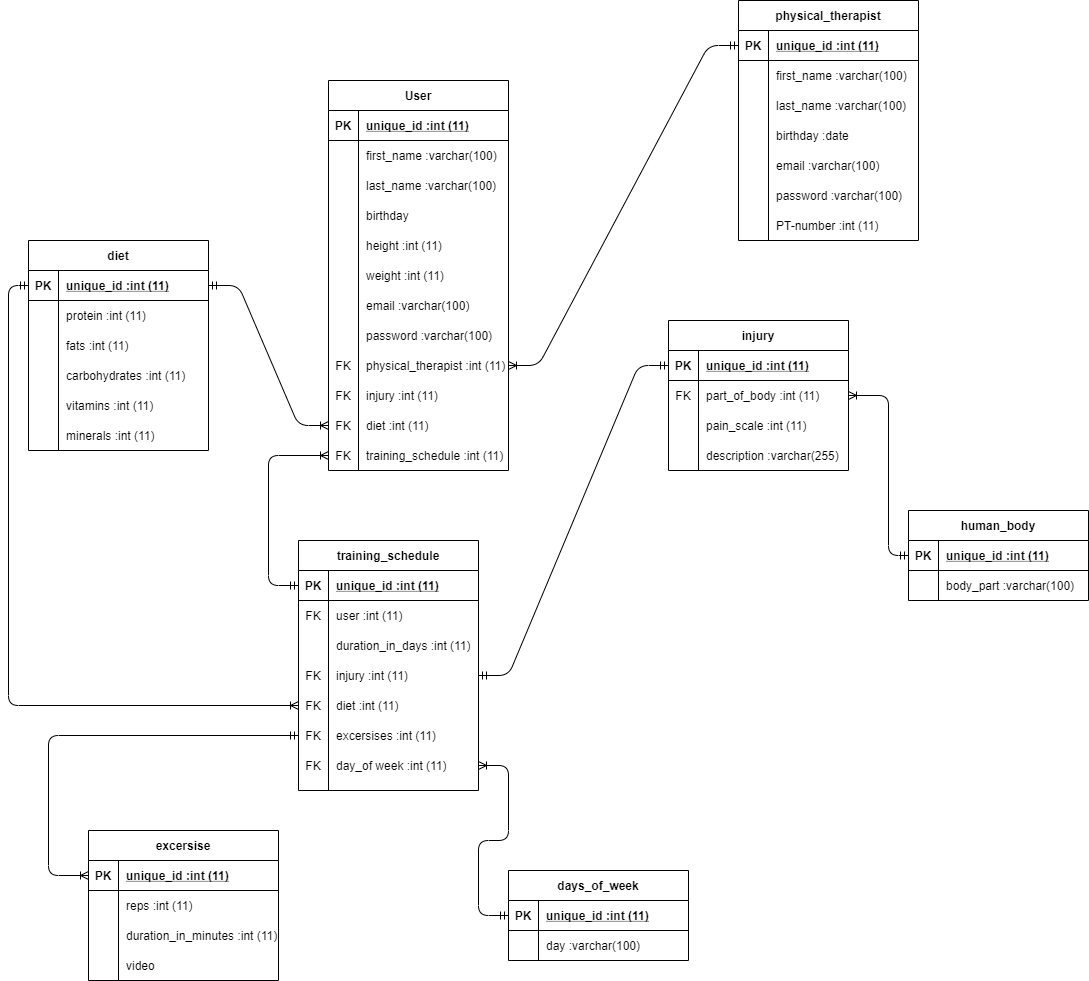
|  |  |
| --- | --- |
| Name | UC05: Selecting the injury |
| Recap | A user selects their injury on the homepage |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. UC02. 2. User selects injury on homepage. 3. Injury is added to database. 4. Injury is added to user |
| Exceptions | 1. No injury is selected. Give error and go back to step 2. |
| Result | Injury is added to a user |

|  |  |
| --- | --- |
| Name | UC06: Adding a training schedule |
| Recap | When an injury is selected, the user must receive a training schedule. |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. UC05. 2. Database sends list with all the exercises for that particular injury. 3. Exercises will be shown in list form. |
| Exceptions |  |
| Result | A list with all the exercises will be shown. |

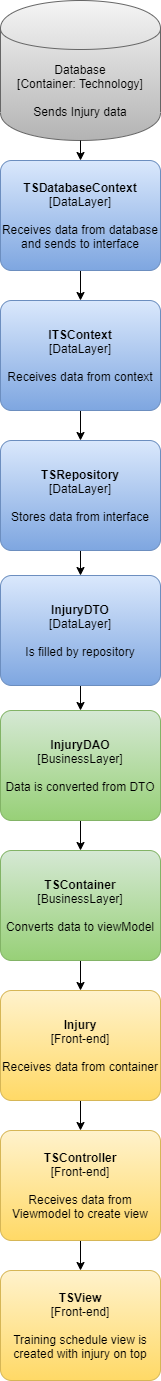
|  |  |
| --- | --- |
| Name | UC07: Chatting with PT |
| Recap | When a PT is added to a user, the user must be able to chat with his PT |
| Actors | User, physical therapist, system, database |
| Assumptions |  |
| Scenario | 1. UC03. 2. User sees chat page. 3. User types and sends message. 4. Message is sent to database. 5. Database sends message to PT view. 6. PT receives message |
| Exceptions | 1. Message input is null. Give error and go back to step 3. |
| Result | Message is sent to PT or back |

|  |  |
| --- | --- |
| Name | UC08: Adding a diet form to user |
| Recap | A PT can add a diet form to a user if needed. |
| Actors | User, system, database |
| Assumptions |  |
| Scenario | 1. User has received training schedule. 2. PT sees training schedule and adds form by adding file. 3. File is sent to database. 4. Database sends file to user view of training schedule. 5. User receives diet form. |
| Exceptions | 1. No file found. Give error and go back to step 2. |
| Result | PT added diet form to user. |

# Database model

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

# Architecture Diagram



1. Database
2. Data-layer
3. Business-layer
4. Front-end