Reha project

Technical Solution Description

Author: Aleksey Maksimov

Content

[Introduction 3](#_Toc56172359)

[Technologies and frameworks 3](#_Toc56172360)

[Database 4](#_Toc56172361)

[Database Description 5](#_Toc56172362)

[User interface 7](#_Toc56172363)

[Additional features 8](#_Toc56172364)

[Services 9](#_Toc56172365)

[Entities 11](#_Toc56172366)

[Screenshots 14](#_Toc56172367)

## Introduction

Reha project it is a web application for medical rehabilitation organization. The application makes it possible to organize the process of treating patients in a medical facility and monitor the course of treatment.

## Technologies and frameworks

* Spring MVC
* Spring security
* Hibernate
* Postgres
* EJB
* JavaScript
* JSP
* JSF
* MQ
* WebServices
* JUnit
* Maven
* WildFly

## Database

## Database Description

**Tables**

|  |  |
| --- | --- |
| **Table** | **Description** |
| patients | Patients information name, birthday, insurance number etc. |
| patient\_diagnozes | Contain information about every patients diagnosis |
| diagnozes | Classifier of Diseases and Related Health Problems |
| assignment | Patients assignments: time pattern, dose of medications, period, etc. |
| treatments | List of treatments available for assignment |
| events | All events which planet, canceled and done for patient |
| employees | Table of employees-users of application |
| roles | Contain name of role and id |

**Relationships**

|  |  |
| --- | --- |
| **Table** | **Description** |
| employee with patients | One to many |
| patients with events |
| patients with assignment |
| assignment with events |
| treatment with assignment |
| patient\_diagnozes | Many to many |
| employees\_roles |

Description of the model

**Employee -** registered user of the service. Medical facility staff. Depends on Role, employee can manage other employees, patients, assignments and events.

**Patient -** contains information about the patient. Also have assignments and diagnoses. Managed by employees with roles ADMIN and DOCTOR

**Assignment -** it is plan of medical treatment. Generate Events. Contains information about treatment.

**Event -** contains information about what and how need to do with patient. Generated when patient get new assignment. Can be changed by employee with role NURSE.

**Diagnose** – medical diagnose, which can be added to the patient by a doctor.

**Treatment** - medical treatment, which can be added to the patient by a doctor.

**Role** - defines the user's role. Depending on the role, the employee has different rights.

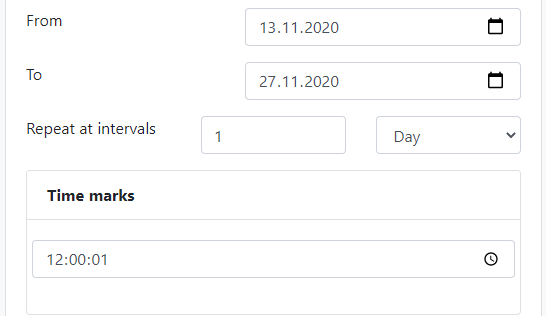
## User interface

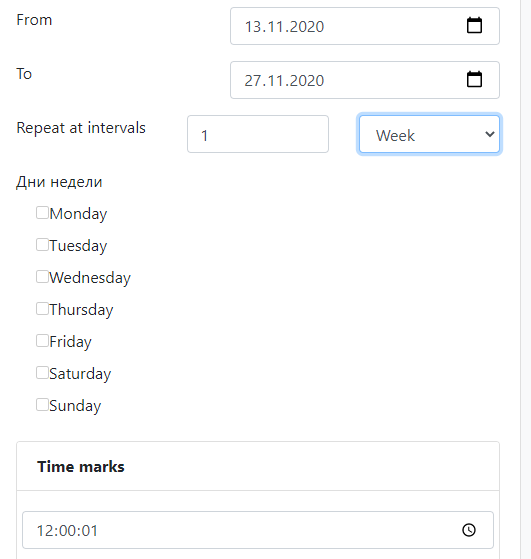
List of available pages

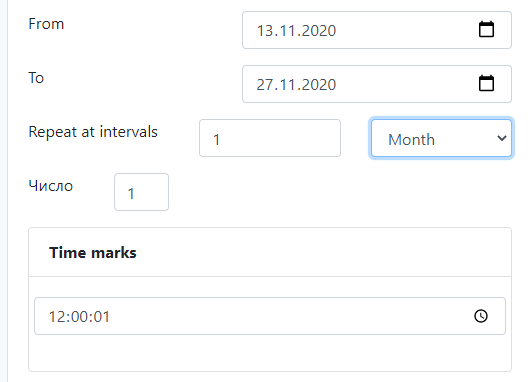
|  |  |  |
| --- | --- | --- |
| **Page** | **Available for** | **Description** |
| Login | All users | Login to application |
| Password | All authenticated users | Page for password change |
| Patients list | ADMIN, DOCTOR | List of patients |
| Patient details | ADMIN, DOCTOR | Full description of the patient (assignments, diagnoses) |
| Events | ADMIN, NURSE | List of events |
| Event | ADMIN, NURSE | Event edition page |
| Users | ADMIN | List of all registered employees |

## Additional features

Appending new assignment to the patient automatically generates all necessary events, considering selected time pattern options. Canceling assignment, automatically canceling all events connected with this assignment.







## Services

|  |  |
| --- | --- |
| **Service** | **Description** |
| Assignment service | Transfers all CRUD operations with entity “Assignment” to repository layer. Convert Assignment entity to/from DTO |
| Diagnoze service | Transfers all CRUD operations with entity “Diagnoze” to repository layer. Convert Diagnoze entity to/from DTO |
| Event service | Transfers all CRUD operations with entity “Event” to repository layer. Convert Event entity to/from DTO |
| Clinic service | Service contains logic of automatic generation and canceling of events |
| PatientDiagnoze service | Transfers all CRUD operations with entity “PatientDiagnoze” to repository layer. Convert PatientDiagnoze entity to/from DTO |
| Patient service | Transfers all CRUD operations with entity “Patient” to repository layer. Convert Patient entity to/from DTO |
| Treatment service | Transfers all CRUD operations with entity “Treatment” to repository layer. Convert Treatment entity to/from DTO |
| User service | Transfers all CRUD operations with entity “User” to repository layer. Convert User entity to/from DTO.  Provide registration and password changing. |

## Entities

Assignment

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| patient |  |
| treatment |  |
| events |  |
| startDate |  |
| finishDate |  |
| timePattern |  |
| intervalSize |  |
| intervalType |  |
| daysOfWeek |  |
| dayOfMonth |  |
| moments |  |
| dose |  |
| status |  |

Diagnoze

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| mkb\_code |  |
| mkb\_name |  |
| patientDiagnozeList |  |

Event

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| assignment |  |
| patient |  |
| timeStamp |  |
| treatment |  |
| dose |  |
| status |  |
| note |  |
| active |  |

Patient

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| name |  |
| birthDate |  |
| insureNum |  |
| assignments |  |
| patientDiagnozes |  |
| patientEvents |  |
| doctor |  |
| healthy |  |

PatientDiagnoze

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| patient |  |
| diagnoze |  |
| active |  |

Treatment

|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| assignments |  |
| name |  |
| drug |  |

User

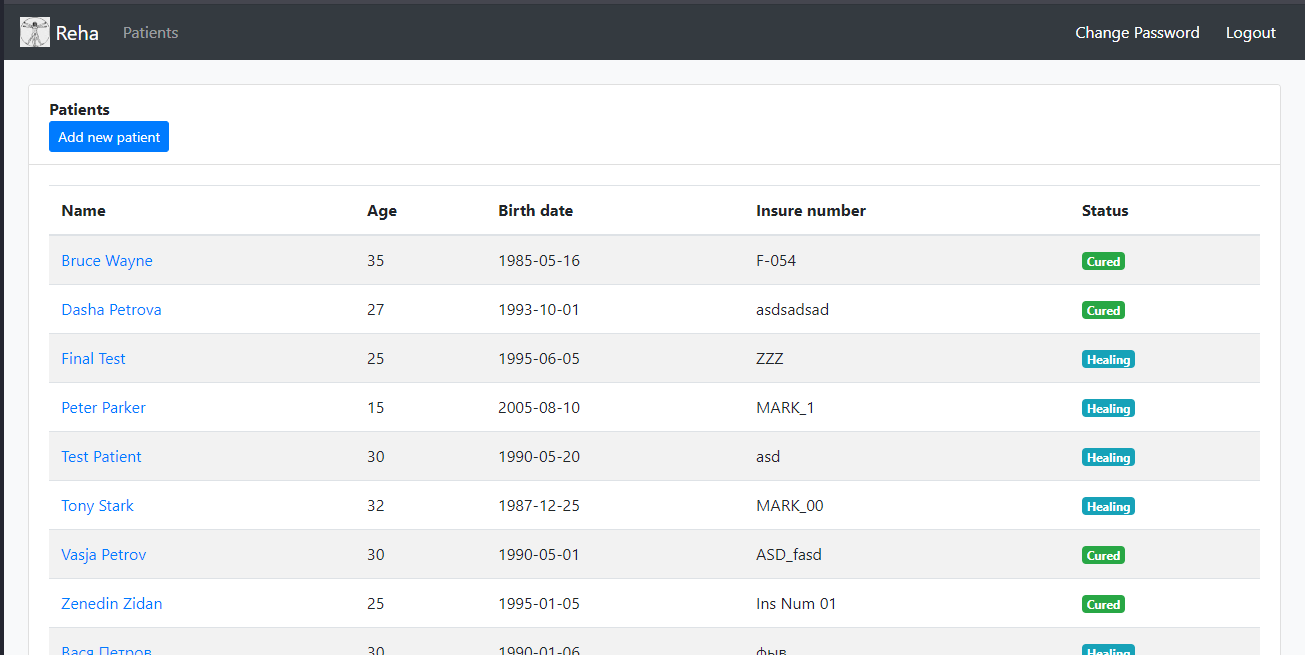
|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| username |  |
| password |  |
| fullName |  |
| roles |  |
| enabled |  |

Role

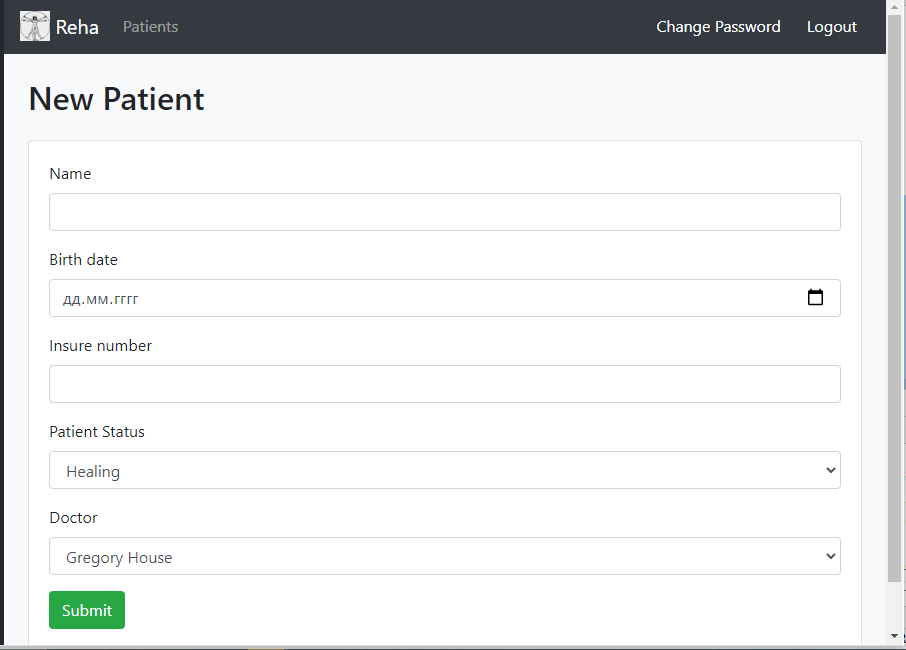
|  |  |
| --- | --- |
| **Field** | **Description** |
| id |  |
| name | Name of role. Must be unique |

## Screenshots

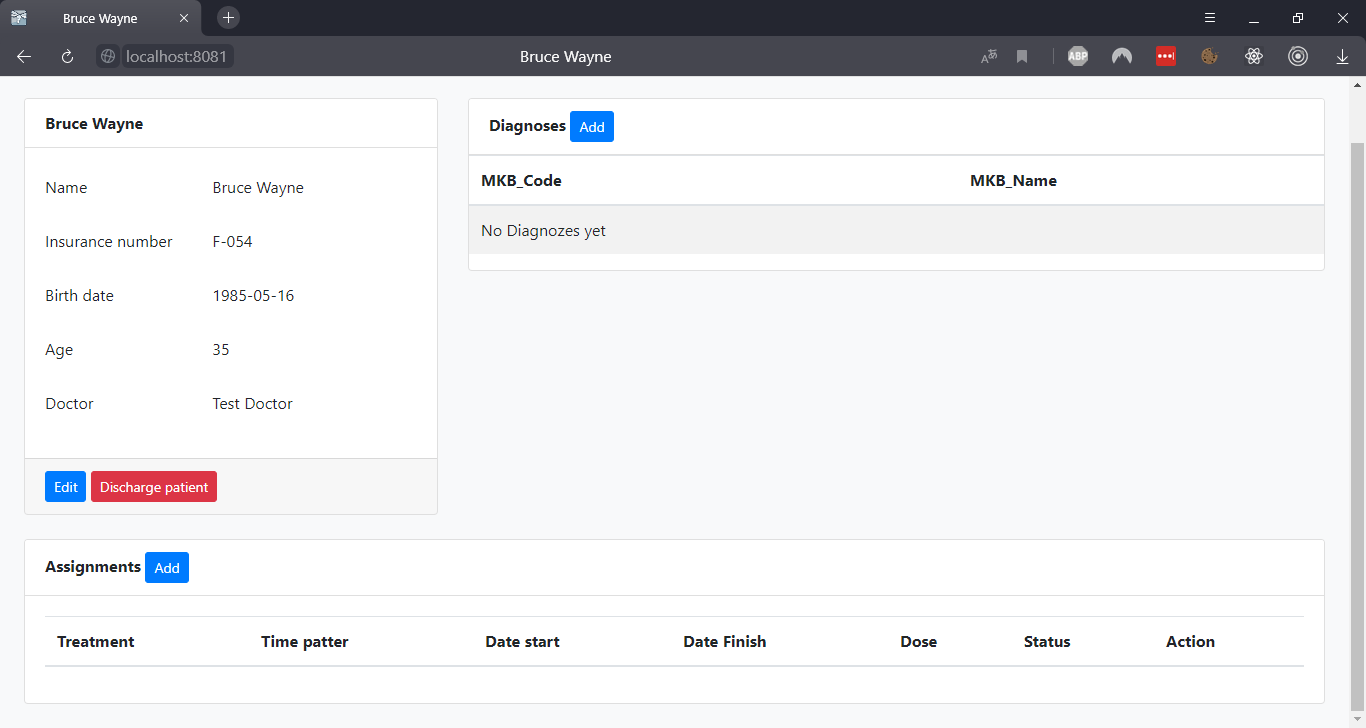
Patients table



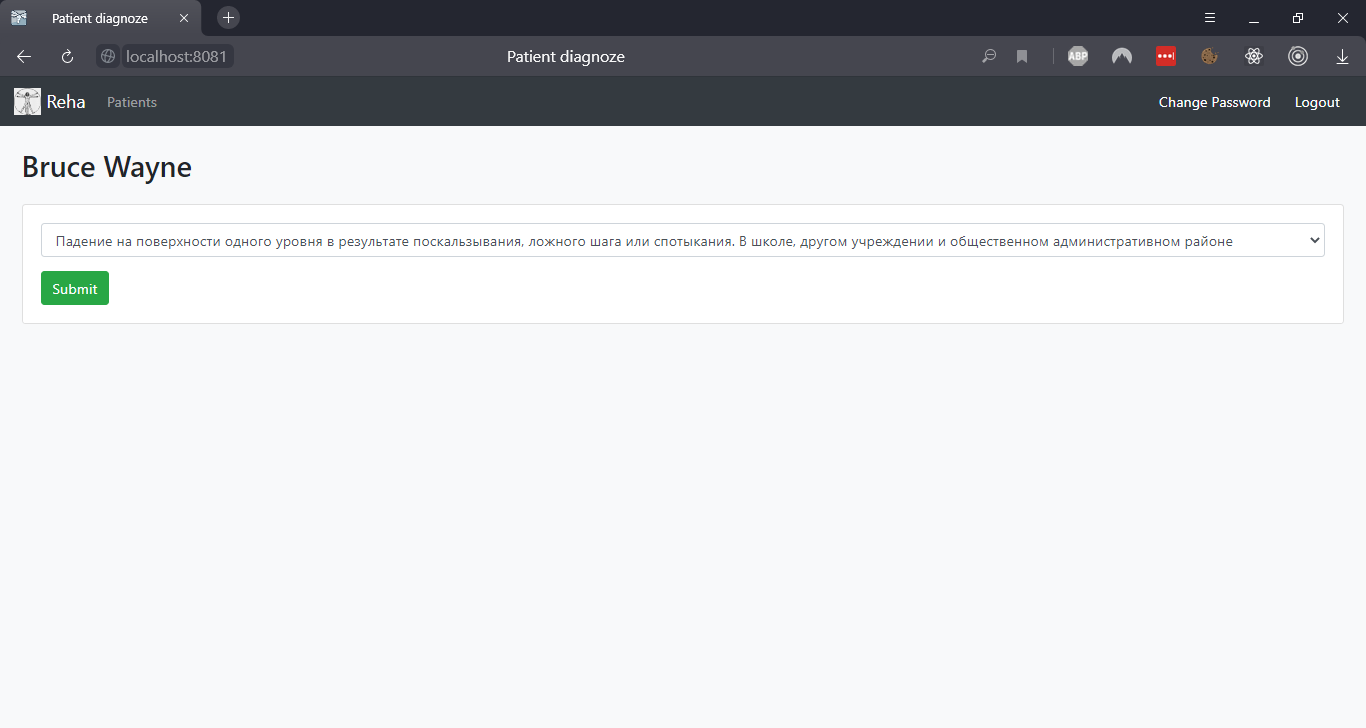
Adding new patient



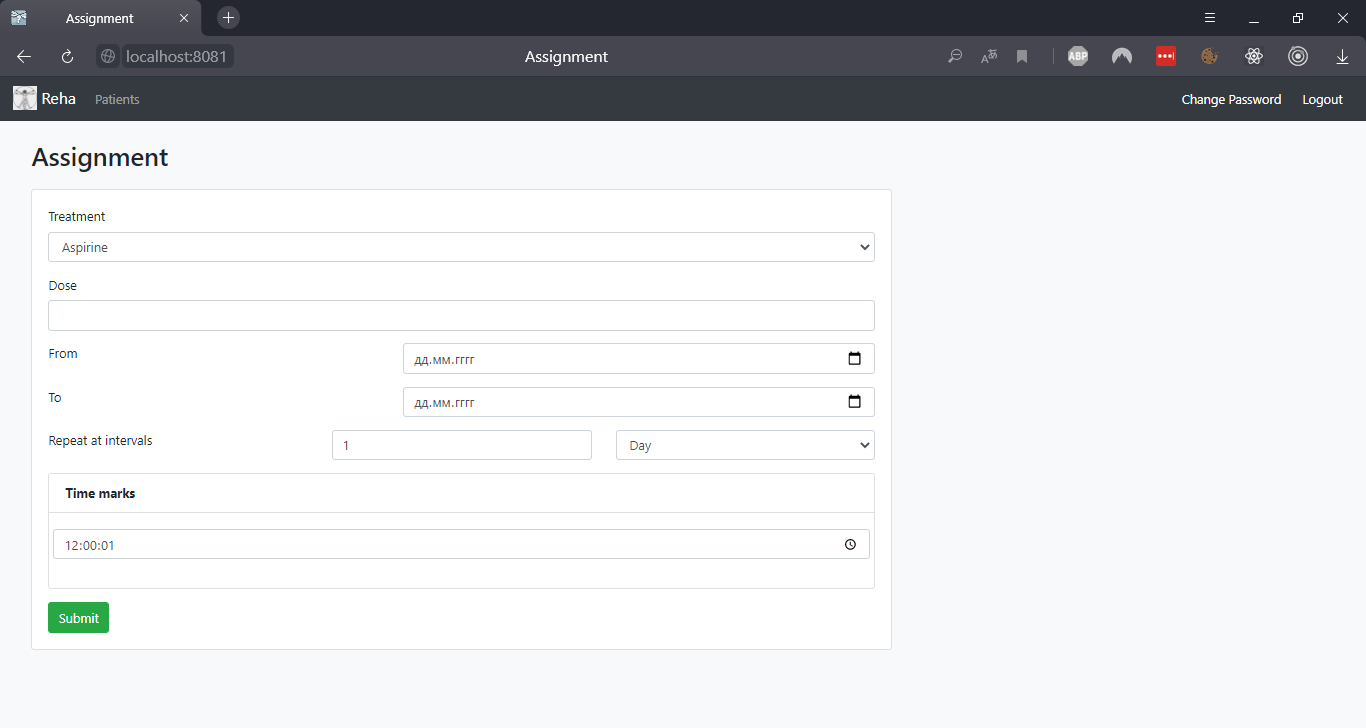
Patient Details



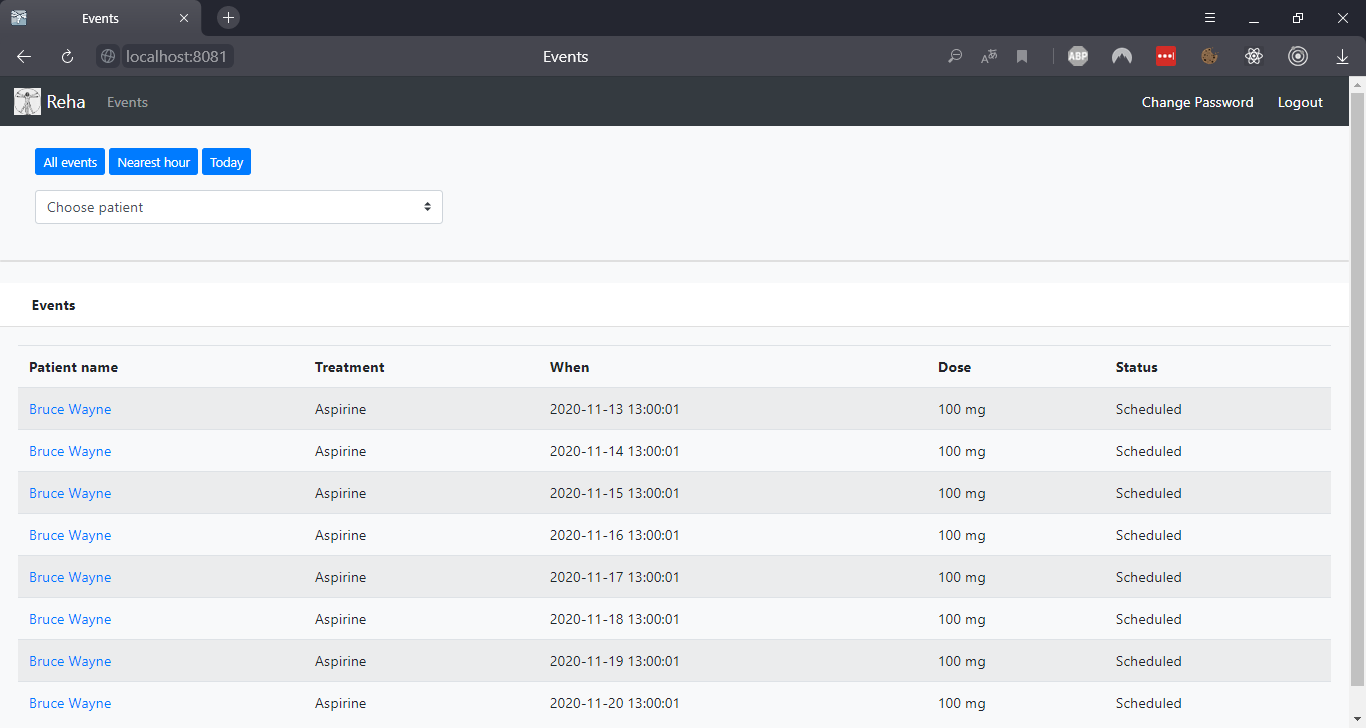
Add new Patients diagnose



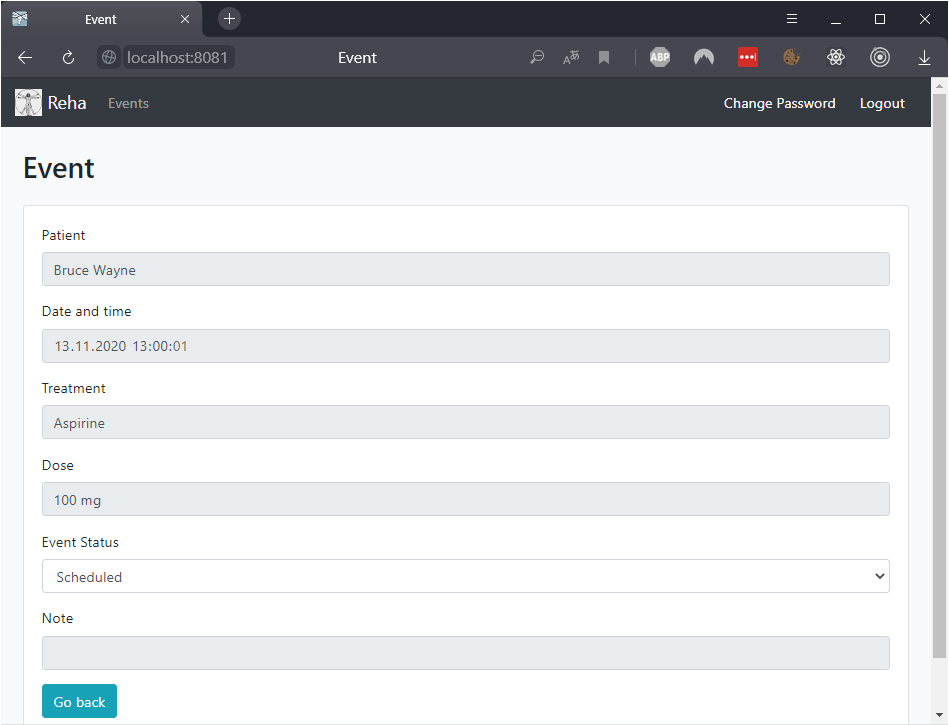
Add new Assignment



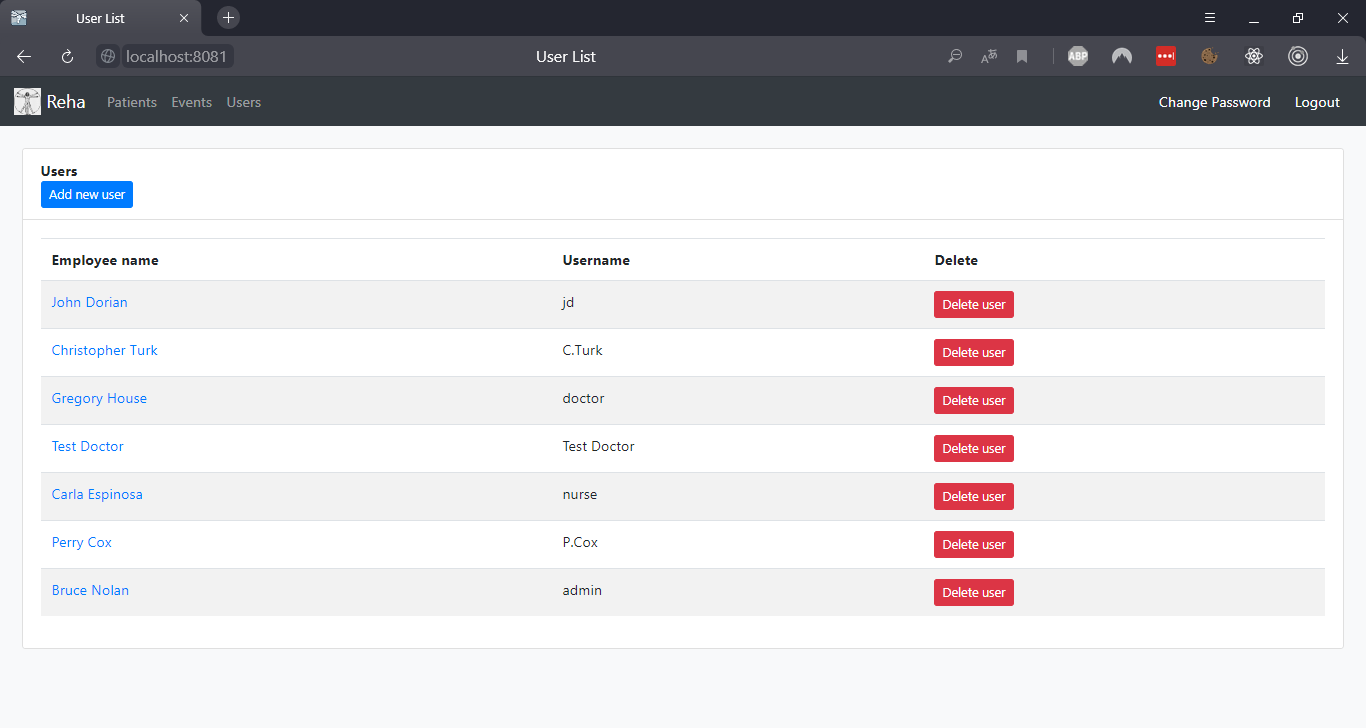
Events list



Event edit



Users list



Add new User

