

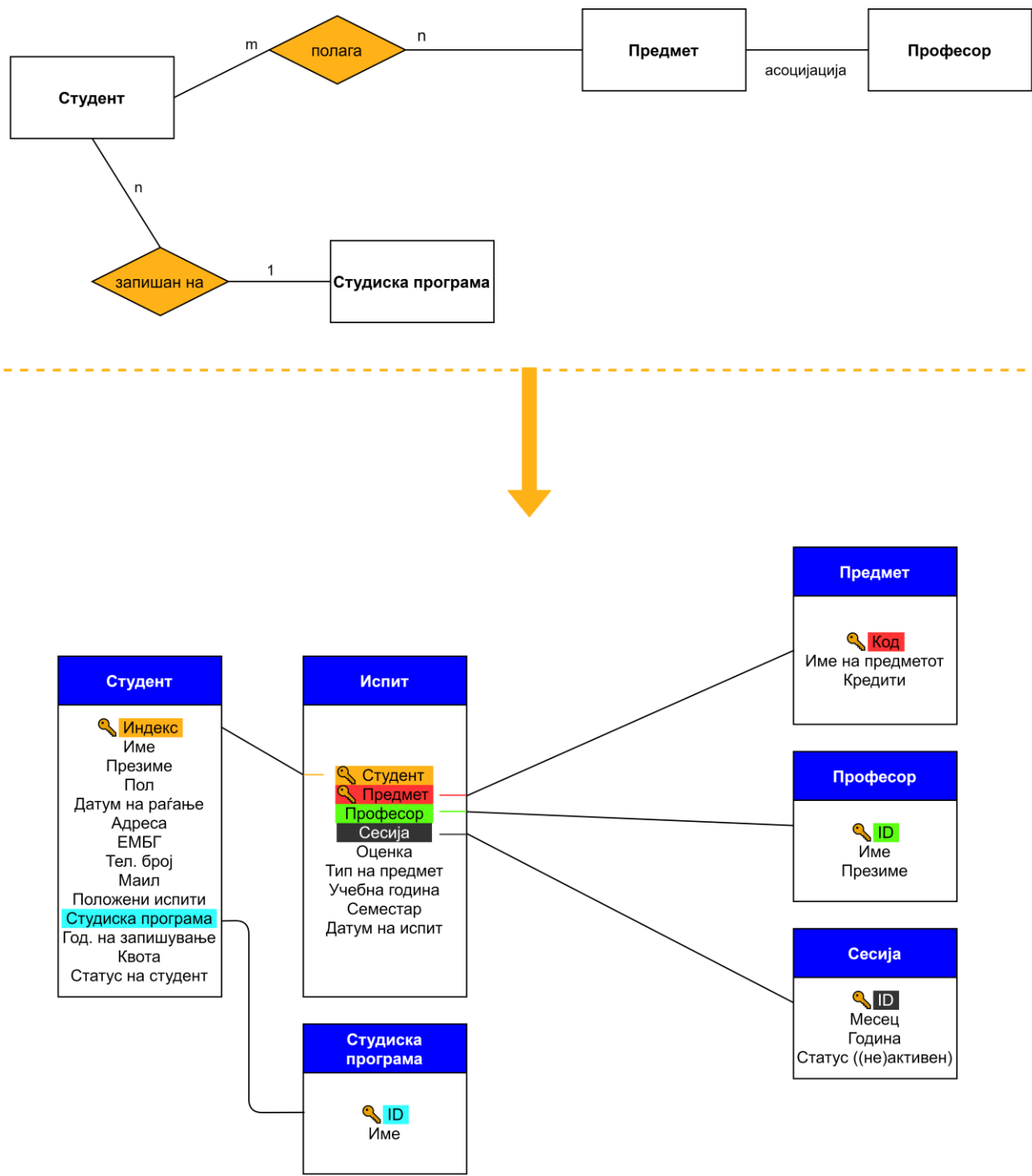
Семинарска работа по предметот  
**Веб Програмирање**

|                |                   |
|----------------|-------------------|
| Студент        | Александар Бојков |
| Индекс         | 173023            |
| Веб апликација | FINKI e-System    |

**Опис:**

Апликацијата овозможува менаџмент на податоците за студентите на Факултетот за Информатички Науки и Компјутерско Инженерство. Постои централна база на податоци. Се чуваат сите детални информации за студентот, како што се името, презимето, бројот на индекс и сл. Секако, главните податоци кои се чуваат за студентот се бројот на положени испити, а и нивни подетални информации (предмет, оценка, професор, датум на полагање итн.). Со оваа апликација е овозможен визуелен интерактивен приказ на сите овие податоци, а секако и нивна манипулација - ова опфаќа манипулација со податоци за студенти (додавање, бришење и менување на студенти), како и манипулација со податоци за испити за секој студент (додавање, бришење и менување на даден испит). Тука секако спаѓаат и податоците за професорите, сесиите и студиските програми.

Апликацијата е поврзана со PostgreSQL база на податоци. Минимизираниот ЕР модел и мапирачката трансформација се во продолжение. Како што беше наведено во правилата за изработка на оваа семинарска работа, имам минимум една врска која е 1:N и минимум една врска која е M:N.



Во склоп на оваа семинарска работа немаше потреба да имплементирам најавување на корисници, затоа што овој систем не е наменет да биде достапен на интернет, туку само во просториите на факултетот од каде ќе може овластено да се пристапува до истиот. Секако, ова решение може да биде проширено така што сето ова може дополнително да се имплементира.

**Опфатеност на концепти, функционалности и искористени технологии и библиотеки:**

- Имплементираните функционалности работат со сигурност и точност од 99%
- HTML, CSS
- Словит дизајн
- Управување со состојби (states)
- Persistence of Data
- Responsive Design
- Bootstrap
- Spring Boot
- ReactJS
- Axios
- JQueryJS
- PostgreSQL DBMS
- JQuery UI (Datepicker/Calendar, Dialog)



**Решение и стартување на апликациите:**

Делот за back-end е изработен како **Spring Boot** апликација (изработена во IntelliJ JetBrains), додека пак делот за front-end е **ReactJS** апликација (изработена во WebStorm JetBrains). Како што напоменах, податоците се чуваат во **PostgreSQL DBMS**.

Back-end делот се наоѓа во фолдерот backend, додека пак front-end делот се наоѓа во фолдерот frontend. Во фолдерот database ги имам експортирано податоците од PostgreSQL базата на податоци користејќи pgAdmin, и сето тоа е зачувано во еден .sql фајл.

За успешно стартување на двете апликации потребно е претходно да се превземат зависностите за истите преку WebStorm и IntelliJ или преку command line со командите npm install и mvn clean install соодветно. Исто така треба да се импортира и базата со податоци која ја имам доставено преку pgAdmin. Со оглед на тоа што не користам JPA, туку моја имплементација на Persistence Layer, нагудувањата за back-end не треба да бидат променети во application.properties, туку во:

```
backend\src\main\java\com\finkisystem\functions\Conn.java
```

и тоа во конструкторот на Conn. Тоа треба да изгледа вака:

```
connection = DriverManager.getConnection("jdbc:postgresql://localhost/<IME_NA_BAZA>",  
                                         "<USERNAME>",  
                                         "<PASSWORD>");
```

**ВАЖНО:** Пред да се стартува back-end делот, поребно е да се поврзе и postgresql драјверот, на следниот начин во IntelliJ:

- File -> Project Structure -> од левата страна се бира Libraries
- Со кликање на + се бира Java библиотека (или само Alt+Insert) и се навигира до патеката каде е драјверот, а тоа е

```
backend\src\main\java\com\finkisystem\functions\postgresql-42.2.12.jar
```

и се клика OK, и пак OK

Двата сервери можат да се стартуваат симултано со IntelliJ и WebStorm. Back-end серверот е поставен на порта 8585, а пак front-end серверот на порта 3000.

Преку command prompt front-end серверот може да се стартува и со командата npm start директно од директориумот frontend.