

# Проект: EMS

Направление: Уеб Приложения

## **Автор:**

Алберт Александров Стефанов

11 “и” клас, НПМГ

[albertas@students.npmg.org](mailto:albertas@students.npmg.org)

## **Ръководител:**

Мирослава Николова

Учител по ИТ, НПМГ

[miroslava.nikolova@npmg.org](mailto:miroslava.nikolova@npmg.org)

# Съдържание

<b>1</b>	<b>Описание на проекта</b>	<b>2</b>
<b>2</b>	<b>Цели и целеви групи</b>	<b>2</b>
2.1	Цели . . . . .	2
2.2	Целеви групи . . . . .	2
<b>3</b>	<b>Основни етапи</b>	<b>2</b>
3.1	Проучване . . . . .	2
3.2	Планиране . . . . .	3
3.3	Имплементация . . . . .	3
<b>4</b>	<b>Ниво на сложност</b>	<b>3</b>
4.1	Възникнали проблеми . . . . .	3
4.2	Разлики в необходимите функции . . . . .	3
<b>5</b>	<b>Логическо и функционално описание</b>	<b>4</b>
5.1	Презентационен слой . . . . .	4
5.2	Бизнес слой . . . . .	4
5.3	Слой за данни . . . . .	5
<b>6</b>	<b>Реализация</b>	<b>5</b>
6.1	Презентационен слой . . . . .	5
6.2	Бизнес слой . . . . .	6
6.3	Слой за данни . . . . .	6

# 1 Описание на проекта

EMS е система за управление на потребителските идентификатори (Identity Management). Идеята е породена от въвеждането на иновативни технологии в училищата — системи за електронно обучение, ученически електронни пощи, безжични мрежи с персонални акаунти, електронни дневници и други.

EMS предлага решение на този проблем. Чрез лесен за използване уеб-базиран панел, отговорните лица — класни ръководители, ЗАТС, директор и заместник-директори, могат да създават потребители във всички системи в зависимост от ролята им (приемно ученик или учител). Всяка позиция идва с ограничения за създаваните видове потребители.

След въвеждане на информацията за даден човек в отделна система, файл или ръчно в EMS, системата създава потребителя спрямо зададения шаблон и го провизира (създава потребителски профил) в отделните системи, зададени първоначално от администратор.

## 2 Цели и целеви групи

### 2.1 Цели

- Интеграция между различни системи
- Автоматизация на провизирането и синхронизацията
- Намаляване на човешките грешки при провизиране

### 2.2 Целеви групи

- Училища и университети
- Организации с голям брой системи и бази данни
- Корпорации с голям брой служители

## 3 Основни етапи

### 3.1 Проучване

1. Проучване на конкурентните продукти

2. Изисквани функции
3. Технологии за имплементацията

## **3.2 Планиране**

1. Приоритизиране на функционалностите
2. Първи системи за изграждане на интеграции
3. Създаване на абстракциите

## **3.3 Имплементация**

1. Създаване на слоя за връзка с бази данни
2. Имплементиране на синхронизатор
3. Създаване на административен панел

# **4 Ниво на сложност**

## **4.1 Възникнали проблеми**

- Липса на cross-platform LDAP библиотеки с пълни функционалности — паралелно с проекта се разработва библиотека на име SimpleLdap
- Голяма разлика в необходимите функции на различните организации

## **4.2 Разлики в необходимите функции**

- Различни конвенции за потребителските имена и електронните пощи — задават се в конфигурацията
- Различни бази данни — създадена е абстракция над конекторите с цел лесна подменяемост
- Необходимите бази данни трябва да се зареждат според конфигурация — имплементацията наподобява plugin система
- Вътрешни разработки на организациите — лесно може да се напише конектор за връзка с вътрешните разработки или да се работи със сурови данни от базата им

## **5 Логическо и функционално описание**

### **5.1 Презентационен слой**

#### **5.1.1 Потребителски панел**

- Управление на потребители
- Управление на потребителски роли
- Създаване и управление на заявки за одобрение на права

#### **5.1.2 Административен панел**

- Създаване на шаблони за потребители
- Дефиниране на технически изисквания към бизнес роли

#### **5.1.3 Web API**

- Извеждане на списък с потребители с цел интеграция в корпоративни сайтове
- Извличане на статистики за репортинг системи на трети лица

#### **5.1.4 Email комуникация**

- Изпращане на молби за одобрение по e-mail
- Одобряване или отхвърляне чрез отговор на писмо
- Не е необходимо служителят да е в корпоративната мрежа, за да управлява заявките за права

### **5.2 Бизнес слой**

#### **5.2.1 Синхронизатор**

- Изгражда връзките между различните бази данни и системи
- При подаване на команда на системата, провизира промените
- Следи за ръчна намеса в промяната на правата и прави отчети за несъответствие на данните между целевите бази

### **5.2.2 Мениджър за управление на права**

- Превръща бизнес ролите в конкретни за всяка система права
- Проследява заявките за одобрение на права
- Дава заявка на синхронизатора да приложи промените във всяка система

## **5.3 Слой за данни**

### **5.3.1 Системи - източници**

- Съдържат въведените данни за потребителите
- Използват се като източник на информация за бизнес ролите и личните данни

### **5.3.2 Целеви системи**

- Съхраняват техническите роли и ги прилагат
- Получават единствено необходимите за функционирането им данни

### **5.3.3 Смесени системи**

- Едновременно са източници и целеви системи
- Може да се използват когато е необходимо обратно да се записват данни, генерирани от други системи

## **6 Реализация**

### **6.1 Презентационен слой**

#### **6.1.1 Потребителски и административен панел**

- Single Page Application, ReactJS
- Responsive Design
- Комуникират с Web API

### **6.1.2 Web API**

- .NET Core Web API
- Предлага RESTful методи, даващи възможност за комуникация със системата
- Конектор — смесена система

### **6.1.3 Email комуникация**

- Email клиент — приема и изпраща писма
- Email генератор — генерира писма, включващи заявките за права
- Конектор — система-източник

## **6.2 Бизнес слой**

### **6.2.1 Синхронизатор**

- Обработка заявките, насочени към определени системи

### **6.2.2 Мениджър за управление на права**

- 

## **6.3 Слой за данни**

### **6.3.1 Системи - източници**

- Имплементират интерфейс `IDataSource`
- Използват се като източник на информация за бизнес ролите и личните данни

### **6.3.2 Целеви системи**

- Имплементират интерфейс `IDataTarget`
- Получават единствено необходимите за функционирането им данни

### **6.3.3 Смесени системи**

- Имплементират интерфейс `IDataContext`, който включва възможностите както на `IDataSource`, така и на `IDataTarget`
- Може и да се четат, и да се записват данни