

EDVENTAGE

Iteration-3

Indie Hounds

Product Owner

Итерация едно - модел

- Създаване на UML диаграми - за визуално представяне на системата
- Създаване на прототип
- Измисляне на временен дизайн - за да може екипът по-лесно да си представи как трябва да изглежда дизайнът. Под временен се има на предвид, че има възможност дизайнът да бъде променен в процеса на разработка в зависимост

Итерация две - база от данни

- Създаване на базата от данни - база от данни, включваща ученици, преподаватели и админи
- Попълване на базата от данни - попълване на базата за определеният университет
- Свързване на базата от данни - свързване на базата от данни със порталът

Итерация три - основни функционалности

- Създаване на учителски и ученически акаунт - сайтът трябва да предлага възможност за добавяне на нови преподаватели и студенти
- Добавяне и изтриване дисциплина - в динамичният, в който живеем постоянно се развиват едни науки, а други залязват. Поради тази причина трябва да има опция за добавяне на нови дисциплини и изтриване на стари.
- Записване на дисциплина - когато студент реши да запише дисциплина, той трябва да може да бъде добавен в нея

Итерация четири - завършване на проекта

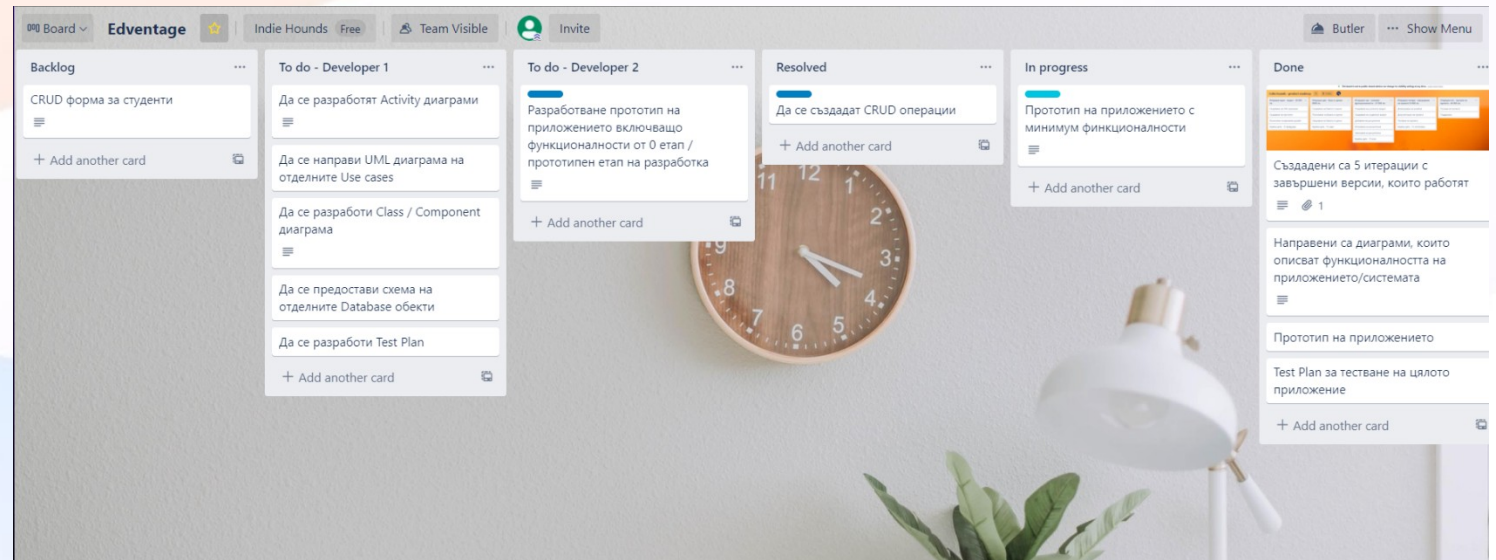
- Довършване на дизайна - след довършване на итерация три вече трябва да се направи завършен вариант на дизайнът на сайта
- Документация на проекта - в нея се включва всякаква информация, която трябва да се знае преди закупуването на проекта
- Тестване на проекта - тестване на проекта за възможни бъгове, които може да попречат на неговата функционалност или дизайн

Итерация пет - пускане на проекта

- Пускане на проекта - след успешното завършване на сайтът и неговото закупуване, то той е готов за използване от университета
- Поддръжка - поддръжка на сайта, в случай че възникне проблем или се желае да бъде подобрен в бъдеще

Project Manager

- С помощта на Trello бяха направени няколко колони, визуализиращи процесите по SCRUM методология.
- Backlog
- To do – Developer 1
- To do – Developer 2
- Resolved
- In progress
- Done

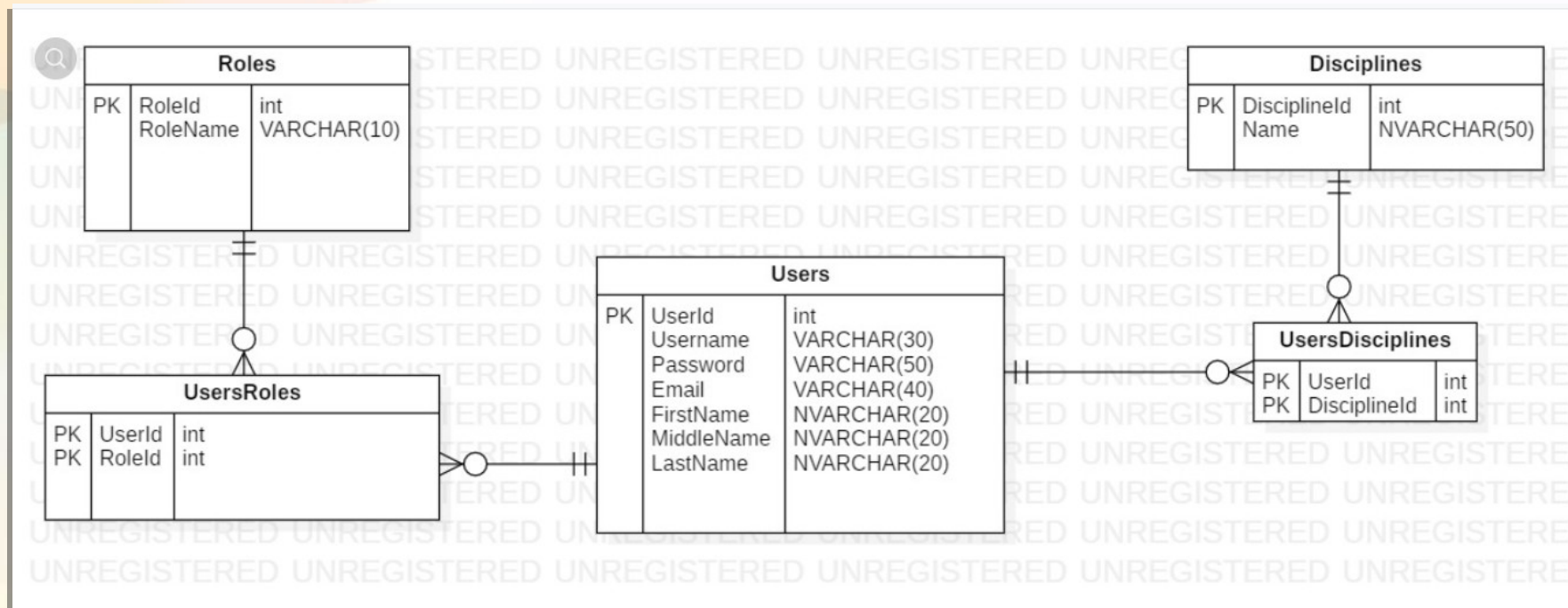


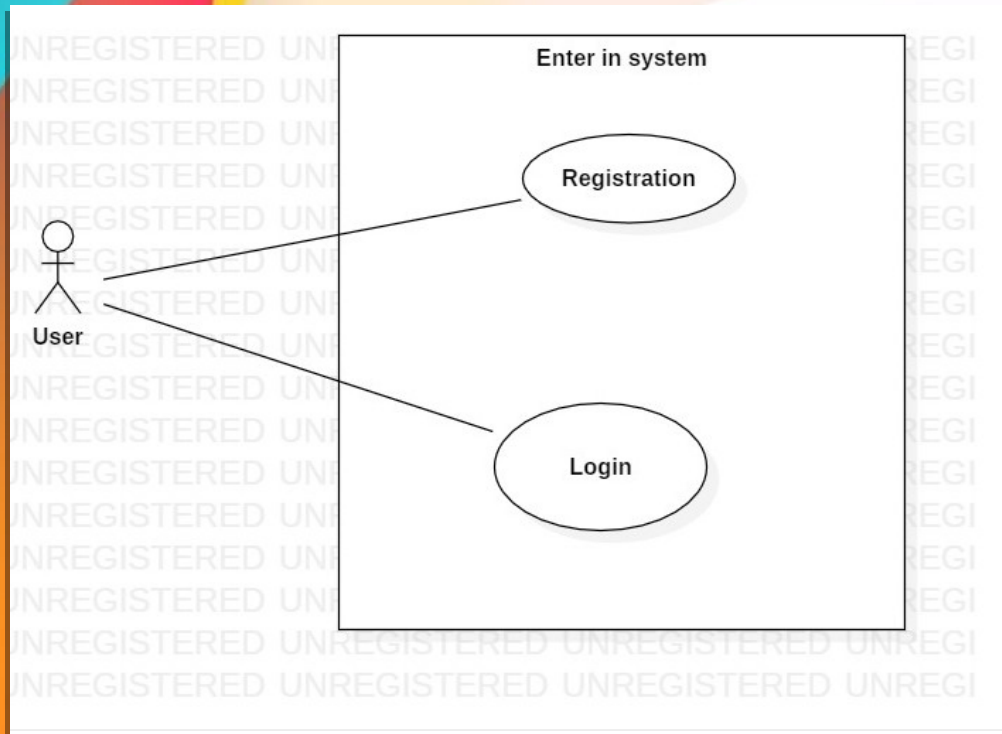
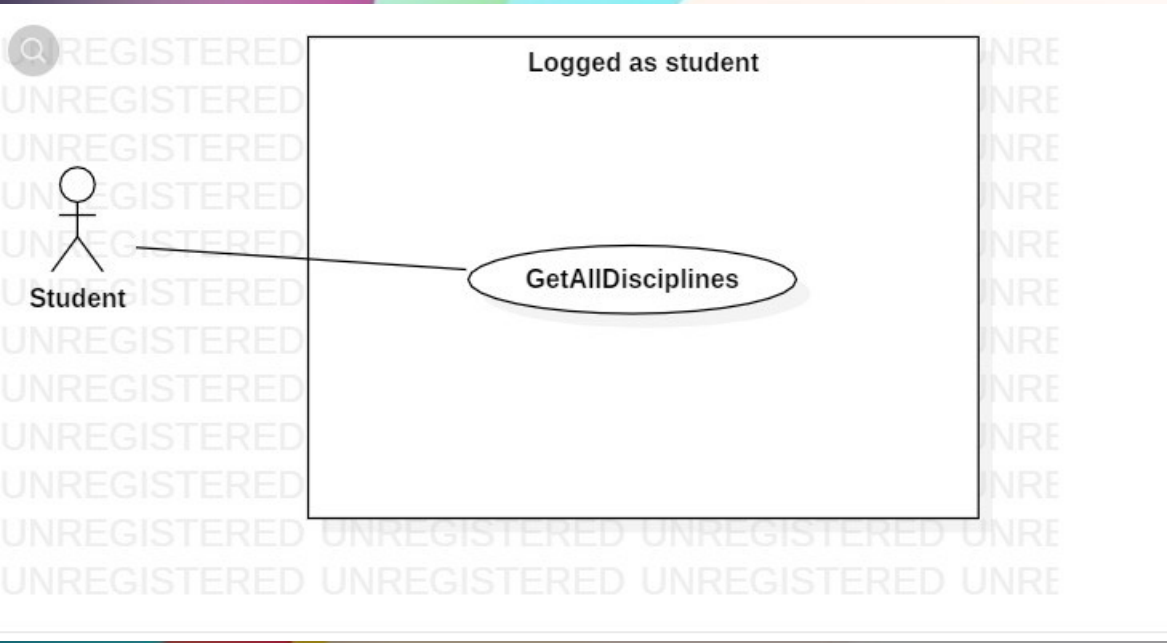
Линк към Trello: <https://trello.com/b/Glv9jGfo/edventage>

Developer - 1

Преди да се направи самият backend, са нужни диаграми.

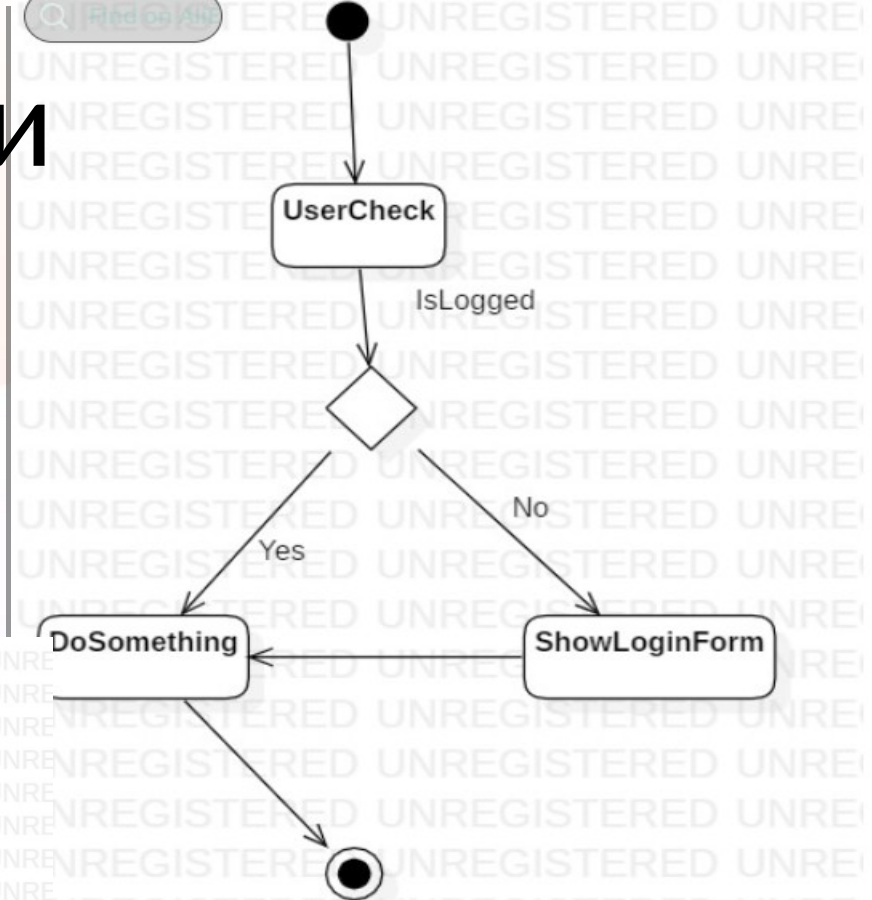
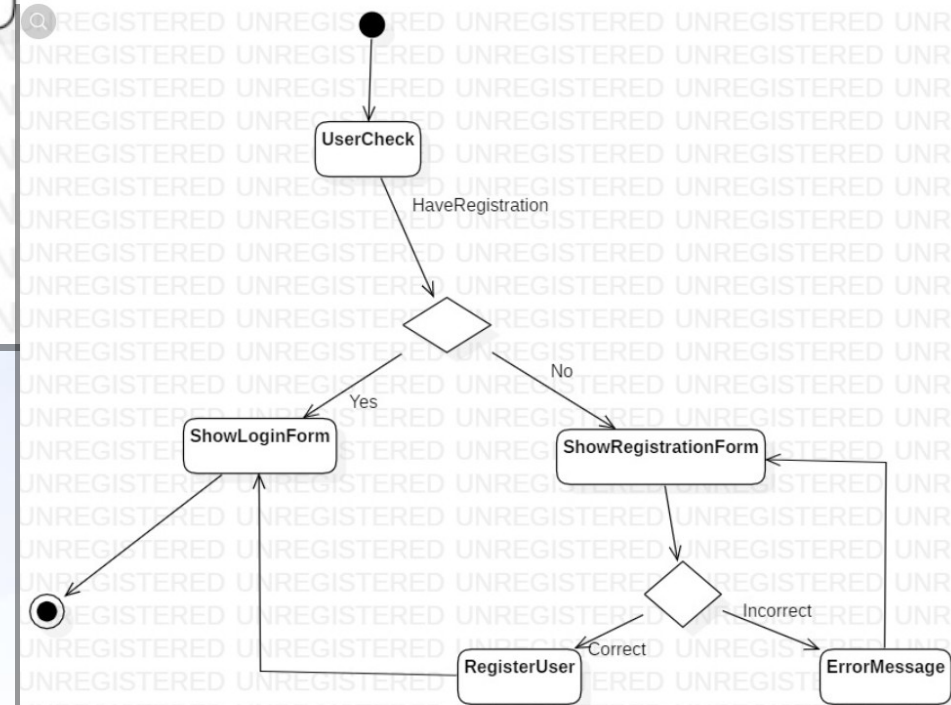
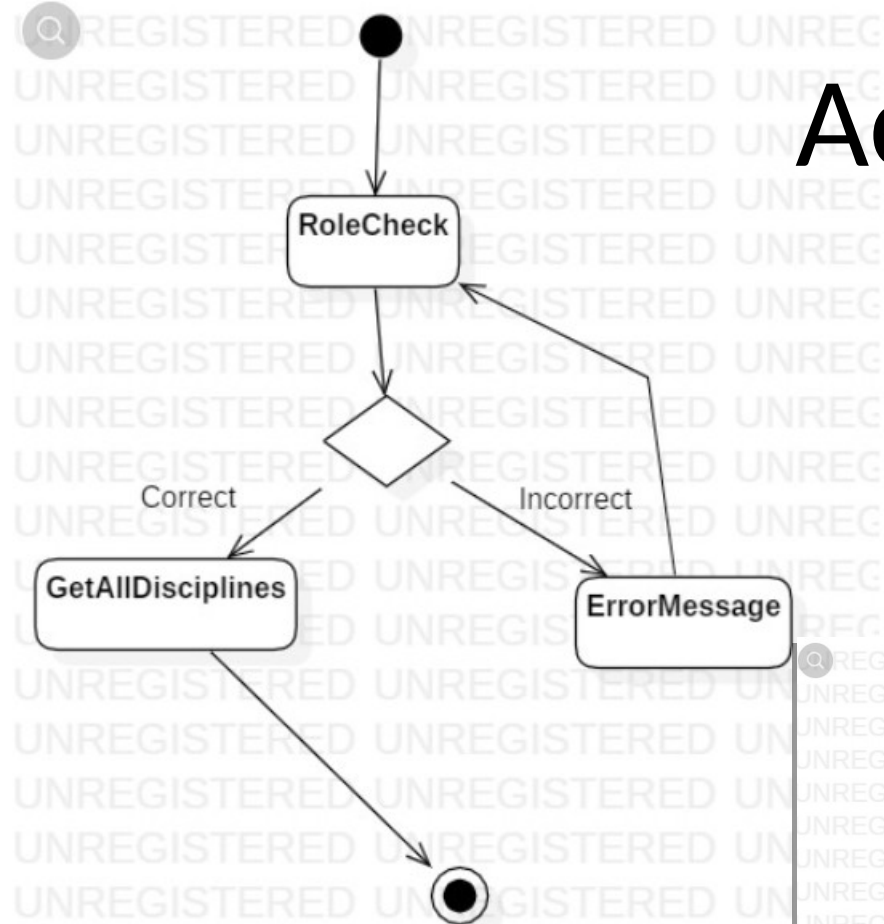
Database диаграма





Use Case диаграмми

Activity diagrams



Developer - 2

```
namespace Edventure.Controllers
{
    public class AuthenticationController : ApiController
    {
        //Authenticate user by id * api/Authentication {json}
        public AuthOUT Get(AuthIN model)
        {
            AuthOUT result = new AuthOUT();

            if(!model.validate())
            {
                result.Error = model.getError()
            }
            else
            {
                User user = new User(model.getUserId());

                result.isLoggedIn = true;
                result.userType = user.getUserType();
                result.userId = user.Id;
            }

            return result;
        }
    }
}
```

```
// GET all courses: api/Courses {json}
public CsOUT Get(CsIN model)
{
    CsOUT result = new CsOUT();

    if(!model.validate())
    {
        result.Error = model.getError();
    }
    else
    {
        //load courses based on user
        result.setCourses(model.UserId);
    }

    return result;
}

// POST: enroll to a course
public CommonOUT Post(EnrIN model)
{
    CommonOUT result = new CommonOUT();

    if (!model.validate())
    {
        result.Error = model.getError();
    }
    else
    {
        result.Success = true;
    }

    return result;
}
```

```

// Create new course : api/Courses {json}
public CommonOUT Put(CrIN model)
{
    CommonOUT result = new CommonOUT();

    if (!model.validate())
    {
        result.Error = model.getError();
    }
    else
    {
        result.Success = true;
    }

    return result;
}

// Delete a course: api/Courses {json}
public CommonOUT Delete(DeIN model)
{
    CommonOUT result = new CommonOUT();

    if (!model.validate())
    {
        result.Error = model.getError();
    }
    else
    {
        result.Success = true;
    }

    return result;
}
}

```

```

//validate fields(length, empty fields, numeric values)
private bool validateFields()
{
    return true;
}

//check if token is valid
private bool validateToken()
{
    return true;
}

//check credentials on db
private bool validateUser()
{
    //get user id from db for further use (if user exists)
    _userId = 1;

    return true;
}

public int getUserId()
{
    if (_userId > 0)
        return _userId;
    else
        //get user id when user is not logged in
        throw new Exception();
}

public string getError()
{
    return _errorMessage;
}
}

```

AuthIN


```
namespace Edventure.Models.Authentication
{
    public class AuthOUT
    {
        public bool isLoggedIn; // true if everything is ok
        public int userId;
        public int userType; // 1 - Admin, 2 - Lecturer, 3 - Student
        public string Error; // empty is no error is present
    }
}
```

Quality Assurance

- Описани сценарии за ръчно тестване на приложението
 - Специален фокус върху тестовете за сигурност на личните данни на потребители
 - Изисквания на приложната област
 - Определяне на допустими стойности за всички полета за въвеждане на информация
 - Тестване както на функционалност, така и на нефункционални ИЗИСКВАНИЯ
- Примерно описание на тестови сценарий:

Като не регистриран потребител ние ще се озовем в началната страница за регистрация

Въвеждаме в полето с етикет за име, масив от символи в коректни размери

Натискаме бутона за регистрация

Резултат: Точно съобщение за грешка: „Необходима е парола за регистрация на потребител!“.

Тестване чрез автоматизация

Използване на Selenium Web Driver.

Основни предимства:

- Поддържа всички масово използвани браузъри.
- Това е инструмент, който е с отворен код, позволява по-добре да се разберът и приложат функционалностите му
 - Има поддръжка за всички големи операционни системи
 - Гъвкав и постоянно обновяван
 - Лесен за използване

Екип:

- Мариян Соколов – Product Owner
- Евелина Генчева – Project Manager
- Борислав Благоев – Developer 1
- Енгин Мустафа – Developer 2
- Еньо Енев – QA